YAŞAR UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

MASTER THESIS

A STUDY ON UNIFORM PARALLEL MACHINE SCHEDULING WITH SEQUENCE DEPENDENT SETUP TIMES

BESTE YILDIZ

THESIS ADVISOR: ASSOC.PROF. AYHAN ÖZGÜR TOY
CO-ADVISOR: PROF.DR. LEVENT KANDİLLER

INDUSTRIAL ENGINEERING

PRESENTATION DATE: 19.01.2022

BORNOVA / İZMİR
JANUARY 2022

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

Jury Members:	**Signature:**
Assoc. Prof. Ayhan Özgür TOY Yasar University	....
Prof. Dr. Levent KANDİLLER Yasar University	....
Asst. Prof. Adalet ÖNER Yasar University	....
Asst. Prof. Erdinç ÖNER Yasar University	....
Asst. Prof. Zehra DÜZGİT İstanbul Bilgi University	....

Prof. (PhD) Yucel Ozturkoglu

Director of the Graduate School

ABSTRACT

A STUDY ON UNIFORM PARALLEL MACHINE SCHEDULING WITH SEQUENCE DEPENDENT SETUP TIMES

Yıldız, Beste
MSc, Master's in Industrial Engineering with Thesis
Advisor: Assoc. Prof. Ayhan Özgür TOY
Co-Advisor: Prof. Dr. Levent KANDİLLER
January 2022

Scheduling problems are essential for decision-making in many academic disciplines, including operations management, computer science, and information systems. Since many scheduling problems are NP-hard in the strong sense, there is only limited research on exact algorithms and their efficiency. This thesis considers the uniform parallel machine scheduling problem with sequence-dependent setup times to minimize the maximum completion time (makespan). We present an IP formulation, which clearly describes our problem and can be used to obtain optimal solutions for small-sized problems. As our problem is NP-hard, we propose a randomized heuristic with an improvement subroutine. The performance of the proposed heuristic through a computational study was tested with 320 instances. We created these instances using the full factorial design of experiment (DOE) with five different factors. Our computational study indicates that the proposed mathematical model takes 22.88 minutes on average, and the heuristic algorithm achieves these results only in 0.062 minutes. The average solutions obtained with the heuristic have an approximately 4% Gap value for average CPLEX solutions. Also, the contribution of the improvement subroutine step to the overall performance of the heuristic is 73.34%.

keywords: parallel machine scheduling, sequence-dependent setup time, full factorial design, randomized heuristic, uniform machines, total completion times

ÖZ

SIRAYA BAĞIMLI KURULUM SÜRELERİ İLE TEK TİP PARALEL MAKİNE ÇİZELGELEMESİ ÜZERİNE BİR ÇALIŞMA

Yıldız, Beste
Yüksek Lisans, Endüstri Mühendisliği Tezli Yüksek Lisans
Danışman: Doç. Dr. Ayhan Özgür TOY
Yardımcı Danışman: Prof. Dr. Levent KANDİLLER
Ocak 2022

Çizelgeleme problemleri; operasyon yönetimi, bilgisayar bilimi ve bilgi sistemleri dahil olmak üzere birçok akademik disiplinde karar vermek için gereklidir. Çoğu çizelgeleme problemi güçlü anlamda NP-zor olduğundan, kesin algoritmalar ve verimliliklerinin nasıl ölçeklendiği konusunda sınırlı araştırma vardır. Bu çalışmada, maksimum tamamlama süresini en aza indirmek için sıraya bağlı kurulum süreleriyle tek tip paralel makine çizelgeleme problemini ele alıyoruz. Problemimizi açık bir şekilde tanımlayan ve küçük boyutlu problemler için en uygun çözümleri elde etmek için kullanılabilecek bir tam sayılı problem formülasyonu sunuyoruz. Sonrasında, problemimiz NP-zor olduğundan, iyileştirme alt rutini ile rastgele bir buluşsal yöntem öneriyoruz. Hesaplamalı bir çalışma yoluyla önerilen sezgisel yöntemin performansı 320 örnekle test edilmiştir. Bu örnekleri, beş farklı faktörlü deneyin tam faktöriyel tasarımını (DOE) kullanarak oluşturduk. Hesaplamalı çalışmamız, önerilen matematiksel modelin ortalama 22.88 dakika sürdüğünü ve sezgisel algoritmanın bu sonuçları yalnızca 0.062 dakikada elde ettiğini göstermektedir. Sezgisel yöntem sonuçları ile matematiksel model sonuçları karşılaştırıldığında, CPLEX yazılımında yapılan sezgisel yöntem ortalama olarak yaklaşık %4 Gap değerine sahiptir. Ayrıca, iyileştirme adımının sezgisel yöntemin genel performansına katkısı %73,34'tür.

Anahtar Kelimeler: paralel makine çizelgelemesi, sıraya bağlı kurulum süresi, tam-etkenli tasarım, sezgisel yöntem, tek tip makine, toplam tamamlanma süresi

ACKNOWLEDGEMENTS

I would like to express my special appreciation and sincere gratitude to my thesis advisor Assoc. Prof. Ayhan Özgür TOY for his immense knowledge, guidance and patience during this study. He consistently allowed my thesis to be my study and he believed in me to complete my thesis successfully.

I would especially like to thank my thesis co-advisor Prof. Dr. Levent KANDİLLER, for his substantial guidance during this way. He always shared his ideas with me to acquire better quality results and enhance my skills as a researcher. Also, he always stands by me, believes in me and encourages me.

Moreover, I appreciate my jury members Asst. Prof. Adalet ÖNER, Asst. Prof. Erdinç ÖNER and Asst. Prof. Zehra DÜZGİT for their many insightful comments, suggestions and contributions.

I also want to state from the heart thank my mother Mahmure, my sister Büşra, my brother Burak Efe and my friend Burak for supporting me throughout my years of researching and writing my thesis. I undoubtedly could not have done this without their unfailing support and continuous encouragement.

Beste Yıldız

İzmir, 2022

TEXT OF OATH

I declare and honestly confirm that my study, titled "A STUDY ON UNIFORM PARALLEL MACHINE SCHEDULING WITH SEQUENCE DEPENDENT SETUP TIMES" and presented as a Master's Thesis, has been written without applying to any assistance inconsistent with scientific ethics and traditions. I declare, to the best of my knowledge and belief, that all content and ideas drawn directly or indirectly from external sources are indicated in the text and listed in the list of references.

Beste Yıldız

19.01.2022

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SYMBOLS AND ABBREVIATIONS

ABBREVIATIONS:

PMSP	Parallel Machine Scheduling Problem
UPMSP	Uniform Parallel Machine Scheduling Problem
SDSP	Sequence Dependent Setup Time
IP	Integer Programming
P	Identical Machines
R	Unrelated Machines
Q	Uniform Machines

SYMBOLS:

N	Number of jobs to be processed.
M	Number of uniform parallel machines.
i, j	Jobs.
k	Machines.
C_i	Completion time of job *i*.
C_{max}	Minimize the maximum completion time (makespan).
v_k	Processing speed of mackine *k*.
p_i	Processing time for job *i* at the base speed.
p_{ik}	Processing time for job *i* on machine *k*.
s_{ij}	Setup time of job *j* immediately after job *i* at the base speed .
s_{ijk}	Setup time of job *j* immediately after job *i* on machine *k*.
L	A large number.

CHAPTER 1
INTRODUCTION

This thesis investigates the fundamental properties of a class of scheduling models commonly used in industrial engineering. Unlike most studies that develop extensions to known models, approaches, or techniques, the emphasis here is to gain insight and understanding. As a direct result of our aspirations, much research was needed before finally developing the ideas presented here. This work considers a uniform parallel machine scheduling problem with sequence-dependent setup times to minimize the maximum completion times (makespan). Tens of thousands of papers addressing different scheduling problems have appeared in the literature since the first systematic approach to scheduling problems was undertaken in the mid-1950s. In this way, parallel machine scheduling problems have an important place in the literature among machine scheduling problems. On the contrary, work on the uniform parallel machine scheduling problem with sequence-dependent setup time is quite limited. We aim to add value by shedding light on this point.

Pinedo (2012) described scheduling as a decision-making process of assigning jobs to resources in a particular order to meet one or more objectives. Also, Allahverdi (2015) stated that scheduling problems can be classified based on the number of stages for jobs to be processed, the number of machines in each stage, job processing requirements, setup time or cost requirements, and the performance metrics to be optimized. Scheduling means determining which jobs can be processed by which machines in what order within a certain period for purposes set, such as ensuring that products are delivered to customers when promised, more efficient use of production resources, and minimization of the total completion time in a manufacturing environment. Ying and Liao (2004) mentioned that efficient scheduling is one of the most critical issues in manufacturing and services in today's competitive industrial world. In addition to the industrial field, other areas benefited from scheduling, such as education, agriculture, transportation, or health research.

Behnamian (2015) stated that scheduling problems are first divided into two classes according to the nature of the problem. The first of these classes is deterministic problems in which the processing constraints and parameters can be ascertained with certainty. The second class is the uncertain scheduling problems in which some processing conditions or parameters cannot be determined in advance. In this context, the uncertain scheduling problems are divided into three types, considering the method of definition of uncertainty. The first one is a fuzzy scheduling problem in which the processing conditions and parameters are modeled using fuzzy numbers. The second one is the stochastic scheduling problem that the stochastic variable is used to specify the processing constraints and parameters. The third one is robust scheduling. Robust approaches aim to create solutions that can absorb some level of the unexpected event without rescheduling. Also, all scheduling problems are classified into five parts. These parts are single machine, parallel machine, flow shop, job shop, and lastly, open shop. In our thesis, we focus on parallel machine scheduling problems and describe the detailed information and sub-headings on this subject in the following sections. Figure 1.1. and Figure 1.2. show the classification of scheduling problems.

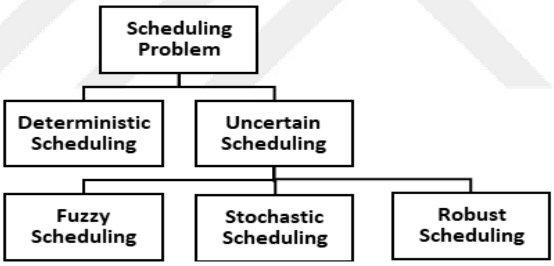

Figure 1.1. A Classification of Scheduling Problems – Part 1

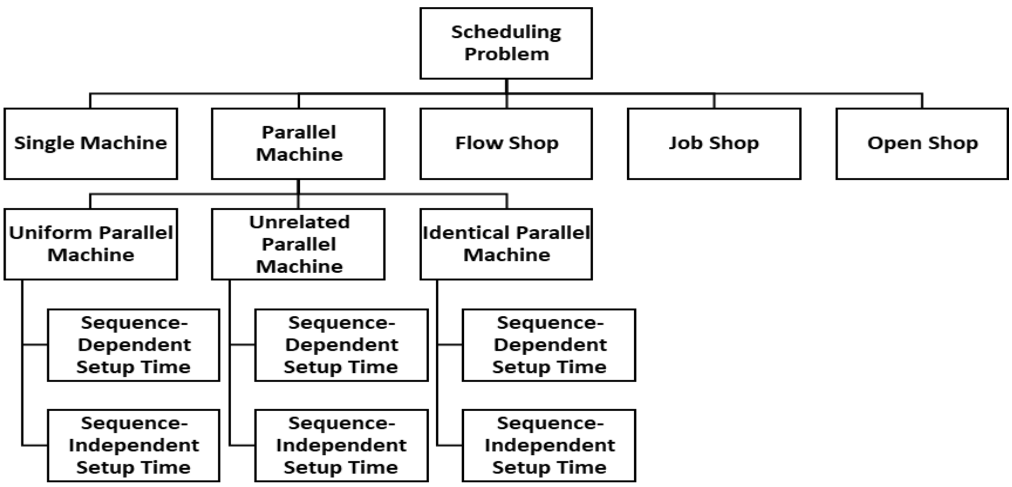

Figure 1.2. A Classification of Scheduling Problems – Part 2

Allahverdi (2015) indicated that in a parallel machine environment, all jobs should be done in a single operation, as in the case of a single machine environment. Also, the operation can be performed by any of *m* machines, which means that *m* machines are running in parallel. In other words, arriving jobs in parallel machine scheduling problems can be processed on any available machines. Each job with different characteristics has a single operation that can be performed on any machine, and job schedules can meet certain criteria based on various performance measures.

Let the number of jobs be denoted by *n*, where the index *i* refers to a job and the number of machines in parallel by *m*, where the index *k* refers to the machines. Each job *i* as to be processed at one of the machines *k* and any machine can do it. Figure 1.3. shows the general representation of this environment.

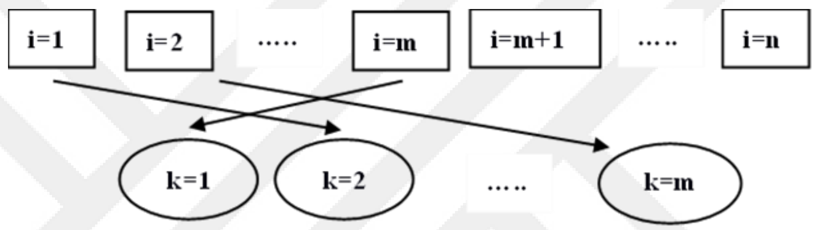

Figure 1.3. The Parallel Machine Environment

The primary work on the parallel machine scheduling problem (PMSP) is by McNaughton (1959) and dates back to the late 1950s. PMSP can be classified into three main categories: (1) identical machines (*P*), where the processing times are the same for all machines, (2) uniform machines (*Q*), where the machines have different speeds but each machine process at a consistent rate, (3) unrelated machines (*R*) where the processing times are arbitrary and have no unique characteristics.

Allahverdi and Soroush (2008) described that setup time is the time it takes to prepare the necessary resource, such as people and machines, required to perform a task, job, or operation. The setup cost is the cost to set up resources before executing a task. Another necessary definition for this thesis is processing time. Processing time is the time required to process a work item. Therefore, the time taken to manufacture a product or provide a service is called processing time. It can be assigned to activities and the entire process. Steps such as reviewing an order, printing shipping labels and packing items, or delivering shipments to a customer can reduce an order's processing time.

Kopanos et al. (2009) pointed out that setup times occur in a large number of industrial and service applications, while a literature review on scheduling problems shows that more than 90 percent of the literature on scheduling problems ignores setup times. Ignoring setup times may be valid for some applications; however, it negatively affects the solution quality of some other scheduling applications. This is because the setup process is not a value-added factor. Hence, setup times need to be clearly considered when planning decisions for industry-critical topics such as increasing efficiency, eliminating waste and improving resource utilization. For the sake of a real-life example of this topic, Loveland et al. (2007) considered the scheduling problem in Dell Inc. They proposed a methodology to minimize the setup cost in the manufacturing system. As a result of this methodology, the production volume was increased by up to 35 percent, and thereby Dell Inc. has saved over $1 million a year.

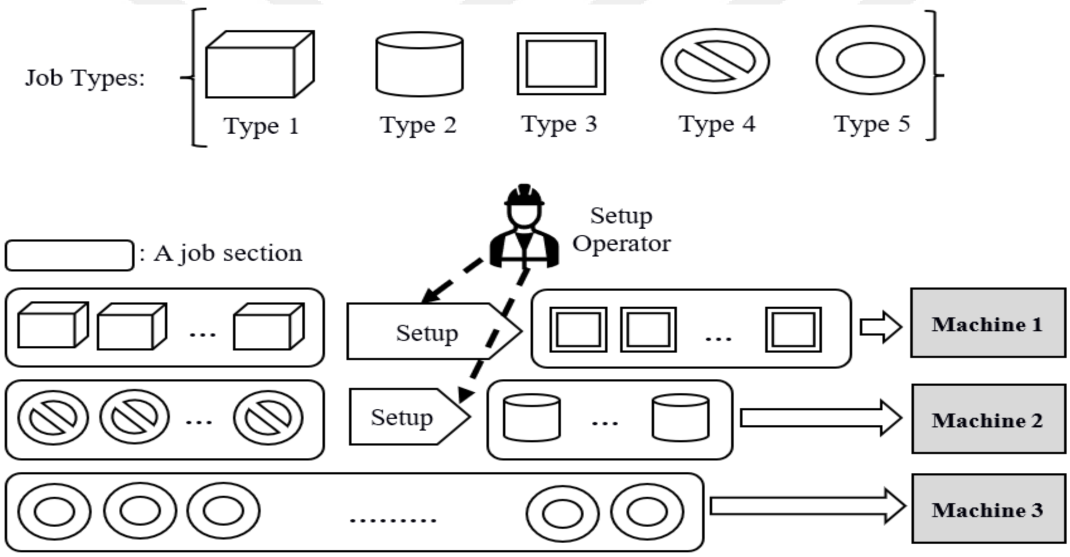

Figure 1.4. Parallel Machine Scheduling with Setup Time Illustration

Figure 1.4. illustrates a simplified example for parallel machine scheduling with setup times. There are five job types and three parallel machines in the system in the example. Jobs are assigned to machines randomly. In this example, jobs of Type 1 and 3 are processed on Machine 1, jobs of Type 2 and 4 are processed on Machine 2, and finally, the job of Type 5 is processed on Machine 3. In each machine, when job types change, a setup is required, and it is performed by a human operator and setup times are different.

Allahverdi et al. (1999) showed that there are two common types of setup (or changeover) structures in classical scheduling problems: (i) sequence-independent- the setup times are usually added to the jobs' processing times, and (ii) sequence-dependent- the setup times depend not only on the job currently being scheduled but also on the immediate preceding job. To give a real-life example of sequence-dependent setup time, Hsu et al. (2009) observed in one of his studies: In manufacturing clothes, the setup (cleaning) time required to prepare for dyeing a future job may differ depending on the colors of the incoming yarn and the color of the yarn that has just finished dyeing. Because before dyeing the yarn, the machine that processes the yarn to be dyed (dyeing tank) must be cleaned. If the previous job is black and the next job is white, the dyeing tank needs to be cleaned completely. On the other hand, if the previous job is white and the next one is black, the dyeing tank needs to be cleaned roughly. Because it is much easier in the system to switch from a light color to dark color; therefore, it requires less setup (cleaning) time when the tank is changed from white to black versus black to white. For this reason, if company owners want to reduce the completion time in the textile industry, these color changes are an important constraint for them. They should care about setup times in their production system.

Ahmarofi et al. (2017) stated that completion time in the manufacturing sector is needed to produce a product through production processes in sequence. Oyetunji (2009) showed that several performance measures are used to evaluate the quality of a schedule. Minimization of the maximum completion time (makespan), minimization of tardiness/earliness, and minimization of the total completion time (*TCT*) are the most common criteria for scheduling problems. Garey and Johnson (1979) pointed out that the PMSP with minimizing the makespan with two identical machines is known to be NP-hard; likewise, Tahar et al. (2006) mentioned a more complex problem with m identical parallel machines and sequence-dependent setup times is also NP-hard. Therefore, heuristics algorithms providing near-optimal solutions in a reasonable runtime are advantageous. We refer the reader to Allahverdi (2015), Allahverdi et al. (1999), Allahverdi et al. (2008), and Gedik et al. (2016) for a comprehensive review of literature on solution methods for different types of PMSP.

Graham et al. (1979) presented that a triplet of notations, $\alpha/\beta/\gamma$, commonly describes a scheduling problem. The first field (α) relates to the machine setting. The second field (β) describes the setup information and details of the processing characteristics, containing multiple entries. The third field (γ) defines the performance measure.

Table 1.1. Field Indicators for the Problem Identifier Triplet of Scheduling Problems

α		γ	
Notation	**Description**	**Notation**	**Description**
1	Single machine	C_{max}	Makespan
P	Parallel machines(identical)	E_{max}	Maximum earliness
Q	Parallel machines(uniform)	L_{max}	Maximum lateness
R	Parallel machines(unrelated)	T_{max}	Maximum tardiness
F_m	m-stage flowshop	D_{max}	Maximum delivery time
J	Job shop	TSC	Total setup/changeover cost
FJ	Flexible job shop	TST	Total setup/changeover time
O	Open shop	TNS	Total number of setups
β		TEC	Total energy consumption
Notation	**Description**	ΣF_j	Total flow time
ST_{si}	Sequence-independent setup time	ΣC_j	Total completion time
SC_{sd}	Sequence-dependent setup cost	ΣE_j	Total earliness
ST_{sd}	Sequence-dependent setup time	ΣT_j	Total tardiness
$ST_{si,f}$	Sequence-independent family setup time	ΣU_j	Number of tardy(late)jobs
$SD_{si,f}$	Sequence-dependent family setup time	$\Sigma w_j C_j$	Total weighted completion time
$SC_{sd,f}$	Sequence-dependent family setup cost	$\Sigma w_j F_j$	Total weighted flow time
ST_{psd}	Past-sequence-dependent setup time	$\Sigma w_j U_j$	Weighted number of tardy jobs
$Prec$	Precedence constraints	$\Sigma w_j E_j$	Total weighted earliness
r_j	Non-zero release date (ready times)	$\Sigma w_j T_j$	Total weighted tardiness
d_j	Due date	$\Sigma w_j TNS$	Total weighted setup times
$split$	Job splitting	$\Sigma w_j W_j$	Total weighted waiting time
M_j	Machine eligibility	$\Sigma h(E_j)$	Total earliness penalties
S	Single Server	$\Sigma h(T_j)$	Total tardiness penalties
h_j	Maintenance activities	$TADC$	Total absolute differences incompletion times
res	Resource constraints		

In Table 1.1., we present the values for each field of this triplet we use in the rest of this paper. For example, a single machine scheduling problem to minimize makespan with sequence-dependent setup times will be noted as $1/ST_{sd}/C_{max}$. Also, many different solution methods have been proposed in the literature to solve scheduling problems. Table 1.2. gives the abbreviations of the solution methods used in the literature reviewed in this thesis. The first column of the table provides the short encodings of the solution methods. In the second column, the expansions of these succinct encodings are given. For example, the solution method of the abbreviation given with SA is the Simulated Annealing solution method for scheduling problems in the literature.

Table 1.2. Abbreviations of The Solution Methods of Scheduling Problems

Description of Abbreviations			
ABC	Artificial Bee Colony	*ICA*	Imperialist Competitive Alg.
ACO	Ant Colony Optimization	*IG*	Iterated Greedy Algorithm
AIS	Artificial Immune System	*ILS*	Iterated Local Search
ALNS	Adaptive Large Neighborhood Search	*MA*	Memetic Algorithm
ATCS	Apparent Tardiness Cost with Setups	*MILP*	Mixed Integer Linear Programming
ATCSR	Apparent Tardiness Cost with Setups and Ready Times	*MIP*	Mixed Integer Programming
B&B	Branch-and-Bound	*PSO*	Particle Swarm Optimization
B&P	Branch-and-Price	*RKGA*	Random Key Genetic Alg.
BRKGA	Parallel Biased Random-Key Genetic Algorithm	*RNG*	Random Number Generation
CP	Constraint Programming	*RSA*	Restricted Simulated Annealing
DE	Differential Evolution	*SA*	Simulated Annealing
EDA	Estimation of Distribution Algorithm	*SEA*	Self-Evolution Algorithm
EMA	Electromagnetism-like Alg.	*SOS*	Symbiotic Organisms Search
FA	Firefly Algorithm	*TS*	Tabu Search
GA	Genetic Algorithm	*VND*	Variable Neighborhood Descent
GRASP	Greedy Randomized Search Procedure	*VNS*	Variable Neighborhood Search
IA	Immune Algorithm		

In this study, we address the problem of scheduling *n* jobs on *m* uniform parallel machines with sequence-dependent setup times to minimize the maximum completion time (makespan). To the best of our knowledge, there are few studies in the literature for this problem. In this context, we provide an IP formulation and propose a randomized heuristic with an improvement subroutine to solve the problem. We evaluate the performance of the proposed algorithm through a computational study.

The rest of this thesis is organized as follows: Chapter 2 gives the literature review for the scheduling problems; Chapter 3 defines the problem, introduces the formulation of the mathematical model, and presents the developed randomized heuristic. Results of computational experiments and comparisons are provided in Chapter 4. Chapter 5 gives the conclusion and direction for further research in related fields.

CHAPTER 2
LITERATURE REVIEW

The parallel machines scheduling problem is one of the most challenging classes of the scheduling problem. Many studies have been conducted on various commercial, industrial and academic fields. Cheng and Sin (1990) considered that parallel machine scheduling problems could be roughly classified into three categories: (1) identical parallel machines, (2) unrelated parallel machines, and (3) uniform parallel machines. In our literature review, we first considered general parallel machine scheduling definitions, divided them into these three main classes, and examined them separately.

2.1. Parallel Machines

In this section, we review papers related to our problem. In a parallel machine environment, all the jobs are required to have a single operation, as in the case of a single machine environment. However, the operation can be performed by any m machines, i.e., the m machines are working in parallel. In other words, arriving jobs in parallel machine scheduling problems can be processed on any available machines. PMSP can be classified into three main categories mentioned in the introduction chapter. The m machines may have the same speed, i.e., identical (P); or have different speeds, i.e., uniform (Q); or completely unrelated (R). A summary of the scheduling literature in parallel machine environments is presented in Table 2.1, Table 2.2 and Table 2.3, where the identical, uniform, or unrelated machines are indicated by the letter P, Q, or R in the second column first indices. To summarize the table structure, the first column shows who wrote the paper and its published year. The second column classifies the problem following Graham et al.'s (1979) 's triple taxonomy, which we mentioned in the previous chapter. The paper examined in this column indicates what kind of machine setting, the performance measure, and the setup information and details of the processing characteristics. Finally, the last column gives the solution methodologies of these papers.

2.1.1. Identical Parallel Machine

First, numerous papers address identical parallel machines. Turker and Sel (2011) studied the $P2/ST_{sd}/C_{max}$ problem. GA algorithm is developed using random data sets and setup operations performed by a single server. The optimum results are obtained using a string-based permutation algorithm.

The problem of $P/ST_{sd}/C_{max}$ is addressed by many researchers. Behnamian et al. (2009) presented the hybridization of an ACO, SA with VNS; combining the advantages of these three individual components is the key innovative aspect of the approach. This proposed algorithm stressed the balance between global exploration and local exploitation. Báez et al. (2019) proposed a hybrid algorithm that combines GRASP and VNS as the improvement procedure. The designed algorithm consists of two phases: construction and improvement, performed using a general VNS. Xu et al. (2013) developed a robust (min-max regret) scheduling model for identifying a robust schedule with minimal maximal deviation from the corresponding optimal schedule across all possible job-processing times. These scenarios are specified as closed intervals. Soares and Carvalho (2020) and Beezão et al. (2017) addressed the problem of $P/ST_{sd}/C_{max}$ with tooling constraint in a flexible manufacturing system (FMS). As main contributions, Soares and Carvalho (2020) studied using a parallel biased random-key genetic algorithm (BRKGA) hybridized with local search procedures organized using VND and they published the results for single benchmark instances available in the literature, which will contribute consistently to the future of the study of the problem. Beezão et al. (2017) proposed two mathematical formulations of the problem and an ALNS metaheuristic. The destroy and repair operators exploit the structures of two well-known and related combinatorial optimization problems, namely the PMSP and the job sequencing and tool switching problem on a single machine.

Hamzadayi and Yildiz (2007) considered the $P/ST_{sd}, S/C_{max}$ problem. Motivated by a real-life problem from the textile industry, Hamzadayi and Yildiz (2007) developed a new MILP model. Also, they considered SA and GA-based metaheuristics. After, they compared the performance of the proposed metaheuristic algorithm solution with basic dispatching rules. This is the first time dealing with the static m identical PMSP with a common server and sequence-dependent setup times.

Arbaoui and Yalaoui (2016) and Tahar et al. (2006) presented the problem of $P/ST_{sd}, split/C_{max}$. Arbaoui and Yalaoui (2016) suggested new approach based on the Benders Decomposition, which can optimally solve the examples discussed in the literature.The problem is divided into two parts. The master problem and the subproblems that using a Traveling Salesman Problem (TSP) exact algorithm. Tahar et al. (2006) studied a new method based on LP techniques. They introduced a lower bound to evaluate the performance of their new approach on a large number of randomly generated instances.

Expósito-Izquierdo et al. (2019) considered the $P/ST_{sd}/\sum C_j$ problem. They firstly proposed a VNS metaheuristic algorithm aimed at finding high-quality and diverse solutions ignoring the learning/tiredness. Then, they studied the effects of learning or tiredness on the obtained solutions in a real-world scenario using a multi-agent simulation approach.

Driessel and Mönch, (2009,2011) presented the problem of $P/ST_{sd}, r_j, prec/\sum w_j T_j$. Driessel and Mönch (2009) suggested a VNS approach that can outperform schedules obtained by a list-based scheduling approach using the ATCSR dispatching rule. Driessel and Mönch (2011) is a considerably extended version of the previous paper, containing more results of computational experiments for various VNS schemes.

Kim et al. (2020) developed a MIP model for the problem of $P/ST_{sd}, split/\sum T_j$. They also proposed a novel mathematical model to offer metaheuristic approaches with new solution representation schemes, solution encoding schemes, and decoding methods by utilizing metaheuristics such as the SA and the GA.

Joo and Kim (2012) considered the problem of $P/ST_{sd}, r_j/\sum w_j TNS, T_j, U_j$. First, they presented the MIP model. Since this mathematical model is not tractable for large problems, GA and SEA metaheuristics are applied to improve the solution efficiency. This is the first time that SEA is a new population-based evolutionary metaheuristic.

Ying and Cheng (2010) and Lee et al. (2010) addressed the problem of $P/ST_{sd}, r_j/L_{max}$. Ying and Cheng (2010) presented IG algorithm. Extensive computational experiments reveal that the proposed heuristic is more effective than state-of-the-art algorithms on the same benchmark problem data set. Lee et al. (2010) proposed SA and RSA algorithms that incorporate a restricted search strategy to eliminate non-effect job moves to find the best neighborhood schedule.

Park et al. (2012) analyzed the problem of $P/ST_{sd}, split, t_j, b_j/\sum T_j$. This paper presented heuristic algorithms that consider job splitting and sequence-dependent major/minor setup times. The performance of the proposed heuristics is compared with the split algorithm, which is embedded into the three heuristics as a slack-based heuristic, dynamic scheduling window-based heuristic, and the latest starting time-based heuristic.

Queiroz and Mundim (2019) solved the $P/ST_{sd}/C_{max}, \sum C_j$ problem with a heuristic that was based on the multiobjective VND and can satisfactorily construct the Pareto front. They recommended neighborhood structures with swap, remove and insertion moves. To the best of our knowledge, there is no application of such a heuristic to solving this problem.

Bosman et al. (2019) addressed the problem of $P/ST_{sd}/w_j\, C_j$. The twist is that the jobs assigned to the machine must obey the order of the input sequence, as is the case in multi-server queuing systems. They establish a constant-factor approximation algorithm. Their approach is very different from what has been used for similar scheduling problems without the fixed-order assumption. They also give a quasipolynomial time approximation scheme (QPTAS) for the particular case of unit processing times.

Ozer and Sarac (2019) proposed the problem of $P/ST_{sd}, M_j/w_j\, C_j$. In this study, an identical parallel machine scheduling problem with sequence-dependent setup times, machine eligibility restrictions, and multiple copies of shared resources (IPMSP-SMS) are considered. MIP models and a model-based GA matheuristic are proposed.

Ying (2012) studied the wafer sorting scheduling problem (WSSP), with minimization of total setup time as the primary criterion and minimization of the number of testers used as the secondary criterion with due dates and maximum machine capacity constraints. Given the strongly NP-hard nature of this problem, a simple and effective IG heuristic is presented. Behnamian et al. (2011) considered a min–max multiobjective procedure for a dual-objective; C_{max} and $\sum E_j + T_j$ in due window problems. Several hybrid metaheuristics were proposed for the addressed problem with three unique features: its population-based evolutionary searching ability belonging to ACO, its ability to balance exploration and exploitation belonging to SA, and its local improvement ability belonging to VNS.

Table 2.1. Literature Review for Identical Parallel Machine

References	**Problem**	**Approach**
Turker and Sel (2011)	$P2/ST_{sd}/C_{max}$ *(Identical 2 Machines)*	GA, String based permutation algorithm
Expósito-Izquierdo et al.(2019)	$P/ST_{sd}/\Sigma C_j$ *with learning or tiredness effect*	VNS algorithm
Arbaoui and Yalaoui (2016)	$P/ST_{sd}, split/C_{max}$	Bender's decomposition and TSP exact algorithm
Behnamian et al.(2009)	$P/ST_{sd}/C_{max}$	Hybridization of an ACO, SA with VNS algorithms
Ying and Cheng (2010)	$P/ST_{sd}, r_j/L_{max}$	IG algorithm
Hamzadayi and Yildiz (2007)	$P/ST_{sd}, S/C_{max}$	MILP model - SA and GA metaheuristics
Driessel and Mönch (2009)	$P/ST_{sd}, r_j, prec/\Sigma w_j T_j$	VNS algorithm and ATCSR dispatching rule
Kim et al.(2020)	$P/ST_{sd}, split/\Sigma T_j$	MIP model - SA and GA metaheuristics
Driessel and Mönch (2011)	$P/ST_{sd}, r_j, prec/\Sigma w_j T_j$	VNS algorithm
Park et al.(2012)	$P/ST_{sd}, split, t_j, b_j/\Sigma T_j$	Slack-based heuristic, dynamic scheduling window-based heuristic and latest starting time-based heuristic
Lee et al.(2010)	$P/ST_{sd}, r_j/L_{max}$	SA and RSA algorithms
Joo and Kim (2012)	$P/ST_{sd}, r_j$ $/\Sigma w_j, TNS, T_j, U_j$	MIP model - SA and SEA metaheuristics
Tahar et al.(2006)	$P/ST_{sd}, split/C_{max}$	LP techniques and lower bound
Xu et al. (2013)	$P/ST_{sd}/C_{max}$	Robust min-max regret scheduling model
Soares and Carvalho (2020)	$P/ST_{sd}/C_{max}$ *with tooling constraint*	BRKGA hybridized with local search procedures using VND
Queiroz and Mundim (2019)	$P/ST_{sd}/C_{max}, \Sigma C_j$	Multiobjective VND and Pareto front neighborhood structure
Báez et al. (2019)	$P/ST_{sd}/C_{max}$	GRASP and VNS algorithm
Bosman et al. (2019)	$P/ST_{sd}/w_j\ C_j$	Quening systems and quasipolynomial time approximation scheme (QPTAS)
Beezão et al. (2017)	$P/ST_{sd}/C_{max}$ *with tooling constraint*	Two mathematical formula and ALNS metaheuristic
Ozer and Sarac (2019)	$P/ST_{sd}, M_j/w_j\ C_j$	MIP model - GA matheuristic

2.1.2. Unrelated Parallel Machine

For unrelated parallel machine scheduling, many researchers addressed the problem of $R/ST_{sd}/C_{max}$ in the literature. Wang et al. (2016) developed a Hybrid Estimation of Distribution Algorithm with Iterated Greedy Search (EDA-IG). This is the first study in the literature dealing with the Estimation of Distribution Algorithm (EDA) applied to the UPMSP-SDST. Abreu and Prata (2019) presented a hybrid meta-heuristic based on GA, SA, VND, and path relinking. The proposed algorithm showed competitive results with an innovative hybridization of GA and neighborhood search algorithms, tested in diverse instances of literature. Furthermore, they presented a granite industry case study to solve real-world problems. Ezugwu et al. (2018) improved the SOS algorithm. They used the ILS strategy to combine variable numbers of insertion and swap moves and LPT rules to enhance the solution quality, performance, and speed. This work is the first to apply an SOS metaheuristic algorithm to solve the UPMSP-SDST. Ezugwu and Akutsah (2018) applied Firefly Algorithm (FA), refined with a robust local search solution improvement mechanism. GA, Invasive Weed Optimization (IWO) and ACO metaheuristic algorithms were developed in parallel to verify and measure the effectiveness of the proposed algorithm. Silva et al. (2019) implemented five algorithms to find solutions for UPMSP-SDST. (1) An exact method (2) VNS, which consists of a metaheuristic that uses the concept of neighborhood structures to find better solutions and escape the local optimum. (3) GA, an optimization method based on the natural evolution process. (4), (5) Two heuristics based on the mathematical modeling called Relax-and-Fix (R&F) and Fix-and-Optimize (F&O) were developed. Ezugwu (2019) proposed three different approaches to solve the problem, including An Enhanced Symbiotic Organisms Search (ESOS) algorithm, a Hybrid Symbiotic Organisms Search with Simulated Annealing (HSOSSA) algorithm and an Enhanced Simulated Annealing (ESA) algorithm.

Tozzo et al. (2018) used GA and VNS to solve the problem due to the difference among their characteristics: the GA is classified as a metaheuristic inspired by nature and based on population, whereas the metaheuristic VNS is not inspired by nature and performs a punctual search through several neighboring structures. These peculiarities allow a complete diversification of the resolution method for the same problem. Diana et al. (2015) proposed an immune-inspired algorithm. The initial population was generated through the construction phase of the GRASP. An evaluation function was applied to help the algorithm escape from local optima. VND local search heuristic developed as a somatic hypermutation operator to accelerate the algorithm's convergence. Lin and Ying (2014) presented a Hybrid Artificial Bee Colony (HABC) algorithm to solve the problem. The performance of the proposed algorithm was evaluated by comparing its solutions to state-of-the-art metaheuristic algorithms and a high-performing ABC-based algorithm. Avalos-Rosales et al. (2015) considered a new makespan linearization and several MIP formulations. These formulations outperform the previously published formulations regarding the size of instances and computational time to reach optimal solutions. A metaheuristic algorithm based on a multi-start algorithm and VND was analyzed. Müller et al. (2015) developed a new MIP-based heuristic combining atomic moves such as insertion, rejection, and closure to generate sequences of such atomic movements minimizing the makespan. This heuristic employed a commercial solver to search the neighborhood in a multi-start algorithm. Vallada and Ruiz (2011) addressed the Genetic Algorithm (GA) for the unrelated parallel machine scheduling problem with sequence-dependent setup times with the objective to minimize the makespan. The proposed GA involved a new crossover operator, which includes a limited local search procedure which was very fast. Two versions of the algorithm were obtained after extensive calibrations using the Design of Experiments (DOE) approach. They reviewed, evaluated and compared the proposed algorithm against the best methods known from the literature. Fanjul-Peyro et al. (2019) suggested a new MILP and a mathematical programming-based algorithm. These new models and algorithms are tested and compared in an extensive and comprehensive computational campaign with the existing ones. The performance of two commercial solvers was also compared in the experiments. Gedik et al. (2018) suggested a novel CP model with two customized branching strategies that utilize CP's global constraints, interval decision variables, and domain filtering algorithms. The performance of the model was evaluated with the state-of-art algorithms. Cheng et al. (2020) studied Random Forest (RF) and Random-Forest-based Hybrid Artificial Bee Colony (RF-HABC) metaheuristics. The main objective of this study was to minimize the makespan in an unrelated PMSP with uncertain machine-dependent and job sequence-dependent setup times (MDJSDSTs).

Arbaoui and Yalaoui (2018) and Fanjul-Peyro et al. (2017) addressed the problem of $R/ST_{sd}, res/C_{max}$. Arbaoui and Yalaoui (2018) formulated the problem using a CP model and solved it using the state-of-the-art solver. They compared this model's results against the existing literature approaches on two sets of small and medium instances. Fanjul-Peyro et al. (2017) modeled two integer linear programming models. The first one was previously proposed in the literature, which was the adaptation of an existing formulation (named UPMR-S). The second one was based on the resemblance to strip packing problems. It was an original contribution of this paper and a novel reformulation of the problem inspired by the strip packing model (named UPMR-P).

Hu et al. (2016) considered the $R/ST_{sd}, r_j/C_{max}$ problem. This paper identified a robust schedule by the min-max regret criterion. To the best of our knowledge, PMSP with uncertain processing time, ready time, and mold change consideration have not been studied in the literature. MILP formulation and an exact algorithm were proposed. Also, they developed a modified ABC algorithm to solve large-sized problems. Al-Harkan and Qamhan (2019) studied the problem of $R/ST_{sd}, r_j, res/C_{max}$. In order to find an optimal solution for this problem, a new MILP was presented. Moreover, a two-stage hybrid metaheuristic based on VNS Hybrid and SA (TVNS_SA) was proposed.

Angel Bello et al. (2018) analyzed the $R/ST_{sd}, h_j/C_{max}$ problem. They presented a mathematical formulation for this problem and derived valid inequalities to improve its performance, allowing the model to obtain optimal solutions for small, medium instances. In addition, they designed an efficient metaheuristic algorithm based on the multi-start strategy for solving larger instances.

Afzalirad and Rezaeian (2016) considered the problem of $R/ST_{sd}, r_j, M_j, Prec, res/C_{max}$. They created a new pure integer mathematical modeling formula. They developed two new metaheuristic algorithms, including GA and AIS, to detect optimal or near-optimal solutions. They also set the parameters of these algorithms using the Taguchi method.

Caniyilmaz et al. (2015) examined the problem of of $R/ST_{sd}, M_j/C_{max} + \sum T_j$. This paper used the new neighborhood approach that gives the different machine assignments for every candidate-job sequence. They took advantage of ABC and GA metaheuristics and this integration benefits to evaluate performances of the algorithms with the real-life problem about quilting work center.

Rauchecker and Schryen (2019) solved the of $R/ST_{sd}, M_j/\sum w_j C_j$ problem. This study adapted an exact B&P algorithm to UPMSP-SDST, parallelized the concerted algorithm by implementing a distributed-memory parallelization with a master/worker approach, and conducted prevalent computational experiments modern high performance computing cluster.

Zeidi et al. (2017) addressed the problem of $R/ST_{sd}, r_j, M_j/(\sum \alpha_j E_j + \beta_j T_j , \sum C_j)$. This study introduced the MIP model to formulate the considered multi-criteria problem. They proposed the namely Controlled Elitism Non-Dominated Sorting Genetic Algorithm (CENSGA) solve the model for real-sized applications. Also, to validate its performance, the algorithm was examined under six metric performance measures and compared with a Pareto-Based Algorithm, namely NSGA-II.

Naderi-Beni et al. (2014) developed the problem of $R/ST_{sd}, M_j, r_j/\sum ML_{max} - ML_j), \sum T_j$.In this paper, a Fuzzy Bi-objective Mixed Integer Linear Programming (FBOMILP) model was presented. The proposed model was solved by two meta-heuristic algorithms, namely Fuzzy Multi-Objective Particle Swarm Optimization (FMOPSO) and Fuzzy Non-dominated Sorting Genetic Algorithm (FNSGA-II) for solving large-scale instances.

Lopes and Carvalho (2007) studied the $R/ST_{sd}, M_j, r_j/\sum w_j T_j$ problem. They developed a new B&P optimization algorithm for the general class of PMSP. A new column generation accelerating method termed 'primal box', Dantzig–Wolfe decomposition, and a specific branching variable selection rule that significantly reduces the number of explored nodes were proposed.

Tavakkoli-Moghaddam et al. (2009) solved the $R/ST_{sd}, r_j, Prec/\sum U_j, C_{max}$ problem. They studied a two-level MIP model to minimize bi-objectives. Since solving the large-sized problem in a reasonable computational time or optimization tools was extremely difficult, this paper presented an efficient GA model to solve the bi-objective PMSP.

Safaei et al. (2015) analyzed the problem of $R/ST_{sd}, r_j, Prec/\sum U_j + C_{max}$. They proposed two Multiobjective Genetic Algorithms (MOGA). Random test problems were produced in medium and large-sized to evaluate the proposed algorithms with tight due dates large-sized with tight due dates. The performances of algorithms were evaluated using the concept of Data Envelopment Analysis (DEA), distance method, and some non-dominated solutions.

Bektur and Sarac (2019) used the $R/ST_{sd}, S, M_j/\sum w_j T_j$ problem. A MILP model was developed, and due to the NP-hardness of the problem, TS and SA algorithms were presented. A modified ATCS dispatching rule obtained the initial solutions of the algorithms.

Cota et al. (2019) addressed the problem of $R/ST_{sd}/C_{max}, TEC$. They considered multiobjective extensions of the Adaptive Large Neighborhood Search (ALNS) metaheuristic with Learning Automata (LA). They solved the large-sized test instances by improving the search process. Moreover, They developed two new algorithms: the Mono-Objective ALNS with Learning Automata (MO-ALNS) and the MO-ALNS/D.

Kongsri and Buddhakulsomsiri (2020) considered the $R/ST_{sd}/C_{max} + \sum T_j$ problem. This paper formulated a MIP model for the UPMSP-SDST that total tardiness. A compromise solution was found with a proper weight between the two measures.

Rocha et al. (2008) analyzed the $R/ST_{sd}/C_{max} + \sum w_j T_j$problem. They used Branch and Bound methods and they ensured the solution by using the GRASP metaheuristic as an upper bound. They suggested some test instances and the metaheuristic results for this type of problem compared with two MIP models.

Zeidi and Hosseini (2015) presented the problem of $R/ST_{sd}/\sum e_j * E_j + t_j * T_j$. A new mathematical model was provided for the considered problem, and due to the complexity of the problem, an integrated meta-heuristic algorithm is designed to solve the problem. The proposed algorithm consisted of GA as the basic algorithm and SA method as the local search procedure.

Chen (2009) solved the $R/ST_{sd}/\sum T_j$ problem. An effective heuristic based on a modified ATCS dispatching rule, the SA method and designed improvement procedures were proposed to minimize the total tardiness of this scheduling problem.

Ekici et al. (2019) examined the problem of $R/ST_{sd}/\sum T_j + E_j$ and machine-job compatibility restrictions and workload balance requirements. They studied a wide range of heuristics, including (i) a sequential algorithm, (ii) a TS algorithm, (iii) a random set partitioning approach, and (iv) a novel matheuristic approach utilizing the local intensification and global diversification powers of a TS algorithm. This study was motivated by the production scheduling operations at a television manufacturer, Vestel Electronics.

Paula et al. (2010) addressed the problem of $R/ST_{sd}/\sum w_jT_j$. This work presented a non-delayed relax and cut algorithm based on a Lagrangean Relaxation of a time-indexed formulation of the problem. Also, Lagrangean pure VNS heuristics were developed to obtain approximate solutions.

Chen and Chen (2009) considered the $R/ST_{sd}/\sum w_jU_j$problem. They studied several hybrid metaheuristics. These metaheuristics began with effective initial solution generators to generate initial feasible solutions; then, they improved the initial solutions by an approach that integrates the VND and TS principles.

Table 2.2. Literature Review for Unrelated Parallel Machine

References	Problem	Approach
Hu et al.(2016)	$R/ST_{sd}, r_j/C_{max}$	Robust min-max regret scheduling model - MILP and exact model - ABC algorithm
Al-Harkan and Qamhan (2019)	$R/ST_{sd}, r_j, res/C_{max}$	MILP model - hybrid VNA and SA (TVNS_SA) metaheuristic
Bektur and Sarac (2019)	$R/ST_{sd}, S, M_j/\Sigma w_jT_j$	MILP model - TS and SA algorithms - ATCS dispatching rule
Naderi-Beni et al.(2014)	$R/ST_{sd}, M_j, r_j/\Sigma(ML_{max} - ML_j), \Sigma Tj$	Fuzzy bi-objective MILP (FBOMILP) model - Fuzzy multiobjective particle swarm optimisation (FMOPSO) and Fuzzy non-dominated sorting genetic algorithm (FNSGA-II)
Wang et al.(2016)	$R/ST_{sd}/C_{max}$	Hybrid EDA and IG (EDA_IG) metaheuristic
Abreu and Prata (2019)	$R/ST_{sd}/C_{max}$	Hybrid meta-heuristic based on GA, SA, VND and path relinking
Rauchecker and Schryen (2019)	$R/ST_{sd}, M_j/\Sigma w_jC_j$	B&P algorithm - Distributed-memory parallelization with a master/worker approach
Tozzo et al.(2018)	$R/ST_{sd}/C_{max}$	GA and VNS metaheuristic
Ezugwu et al.(2018)	$R/ST_{sd}/C_{max}$	ILS strategy - SOS metaheuristic - LPT rules
Afzalirad and Rezaeian (2016)	$R/ST_{sd}, r_j, M_j, Prec, res/C_{max}$	Pure integer mathematical model - GA and AIS algorithms

Table 2.2 (cont'd). Literature Review for Unrelated Parallel Machine

References	Problem	Approach
Zeidi and Hosseini (2015)	$R/ST_{sd}/(\Sigma e_j E_j + t_j T_j)$	Mathematical model - GA and SA metaheuristic
Diana et al.(2015)	$R/ST_{sd}/C_{max}$	Immune-inspired algorithm - GRASP and VND algorithm
Lin and Ying (2014)	$R/ST_{sd}/C_{max}$	Hybrid artificial bee colony (HABC) algorithm
Caniyilmaz et al.(2015)	$R/ST_{sd}, M_j/C_{max} + \Sigma T_j$	ABC and GA metaheuristics
Avalos-Rosales et al.(2015)	$R/ST_{sd}/C_{max}$	MIP model - VND algorithm
Ezugwu and Akutsah (2018)	$R/ST_{sd}/C_{max}$	FA, GA and ACO metaheuristics and Invasive weed optimization (IWO)
Müller et al.(2015)	$R/ST_{sd}/C_{max}$	MIP-based heuristic combining atomic moves - Multi-start algorithm
Vallada and Ruiz (2011)	$R/ST_{sd}/C_{max}$	GA - Design of Experiments (DOE) approach
Silva et al.(2019)	$R/ST_{sd}/C_{max}$	Exact algorithm - VNS, GA - Relax-and-Fix (R&F) and Fix-and-Optimize (F&O) heuristics
Paula et al. (2010)	$R/ST_{sd}/\Sigma w_j T_j$	VNS algorithm - Lagrangean relaxation
Rocha et al.(2008)	$R/ST_{sd}/C_{max} + \Sigma w_j T_j$	Two MIP models - B&B algorithm - GRASP metaheuristic
Tavakkoli-Moghaddam et al.(2009)	$R/ST_{sd}, r_j, Prec/\Sigma U_j, C_{max}$	Novel two-level MIP model - GA to solve bi-objective PMSP
Chen (2009)	$R/ST_{sd}/\Sigma T_j$	SA and modified ATCS dispatching rule
Chen and Chen (2009)	$R/ST_{sd}/\Sigma w_j U_j$	VND and TS metaheuristics
Safaei et al.(2015)	$R/ST_{sd}, r_j, Prec/\Sigma U_j + C_{max}$	Multi objective genetic algorithms (MOGA) - Data envelopment analysis (DEA),
Lopes and Carvalho (2007)	$R/ST_{sd}, M_j, r_j/\Sigma w_j T_j$	B&P algorithm - Dantzig-Wolfe decomposition and a specific branching variable selection rule
Zeidi et al.(2017)	$R/ST_{sd}, r_j, M_j/(\Sigma \alpha_j E_j + \beta_j T_j, \Sigma C_j)$	MIP model - Controlled elitism non-dominated sorting genetic algorithm (CENSGA) - Pareto-based algorithm (NSGA-II)

Table 2.2 (cont'd). Literature Review for Unrelated Parallel Machine

References	**Problem**	**Approach**
Kongsri and Buddhakulsomsiri (2020)	$R/ST_{sd}/C_{max} + \Sigma T_j$	MIP model
Cheng et al. (2020)	$R/ST_{sd}/C_{max}$	Random Forest (RF) and Random-Forest-based Hybrid Artificial Bee Colony (RF-HABC)
Cota et al. (2019)	$R/ST_{sd}/C_{max}, TEC$	ALNS metaheuristic with Learning Automata (LA)
Fanjul-Peyro et al. (2019)	$R/ST_{sd}/C_{max}$	MILP and mathematical programming
Angel-Bello et al. (2018)	$R/ST_{sd}, h_j/C_{max}$	Mathematical model - Multi-start algorithm
Arbaoui and Yalaoui (2018)	$R/ST_{sd}, res/C_{max}$	CP model
Fanjul-Peyro et al. (2017)	$R/ST_{sd}, res/C_{max}$	Two integer linear programming problems (resemblance to strip packing problems)
Ezugwu (2019)	$R/ST_{sd}/C_{max}$	Enhanced Symbiotic Organisms Search (ESOS) algorithm, a Hybrid Symbiotic Organisms Search with Simulated Annealing (HSOSSA) algorithm, and an Enhanced Simulated Annealing (ESA) algorithm.
Gedik et al. (2018)	$R/ST_{sd}/C_{max}$	Noval CP model with two customized branching strategies
Ekici et al.(2019)	$R/ST_{sd}/\Sigma T_j + E_j$	TS and sequential algorithm, random set partitioning and novel matheuristic approach

2.1.3. Uniform Parallel Machine

Lastly, some papers considered resources in scheduling uniform parallel machines, Armentano and Franca (2007) addressed the problem of $Q/ST_{sd}/\Sigma T_j$. They proposed GRASP versions that incorporate adaptive memory principles for solving this problem to minimize the total tardiness with respect to job due dates. Initially, they adapted suitable components for any GRASP procedure, namely, a greedy function and neighborhoods together with a candidate list. Then, they examined the use of long-term memory composed of an elite set of high quality and sufficiently distant solutions.

Balakrishnan et al. (1999) studied the problem of $Q/ST_{sd}, r_j/\Sigma e_j E_j + \Sigma t_j T_j$. For this complex problem, they presented a compact mathematical model and described their computational experience in using this model to solve small-sized problems.

Table 2.3. Literature Review for Uniform Parallel Machine

References	**Problem**	**Approach**
Armentano and Franca (2007)	$Q/ST_{sd}/\Sigma T_j$	GRASP and adaptive memory principles
Balakrishnan et al. (1999)	$Q/ST_{sd}, r_j/\Sigma e_j E_j + t_j T_j$	Mathematical model

CHAPTER 3
PROBLEM DESCRIPTION & ANALYSIS

In this thesis, we consider the uniform parallel machine scheduling problem with sequence-dependent setup times, denoted as $Q/ST_{sd}/C_{max}$. In uniform parallel machine scheduling, n jobs are processed on m machines in parallel ($n > m$), where machines have different processing speeds. The processing speed of machine k, ($k = 1, 2, .., m$), is denoted by v_k. For example, if $v_1 = 2v_2$, then Machine 1 processes a job twice as fast as Machine 2. Job i, ($i = 1, 2, ..., n$), has the processing time of p_i at the unit processing speed. Therefore, the processing time of job i on machine k is $p_{ik} = p_i / v_k$. Note that when all machines have identical speeds, i.e., $v_1 = ... = v_m$, the problem we consider herein transforms into the identical parallel machine scheduling problem. Hence the latter is a special case of the problem we consider. A setup is required before processing a job in a machine. We consider the setting where these setup times are sequence-dependent. Similar to the processing times, setup times depend on the machine speeds. Namely, if job j will be processed immediately after job i on the same machine, the setup time is s_{ij} at the unit processing speed. For a particular machine k, the setup time of job j immediately after job i is denoted by $s_{ijk} = s_{ij} / v_k$. The objective is to minimize the maximum completion time (makespan).

Example 1. Suppose that there are eight jobs and three machines in the manufacturing system. All job has a processing time at the base speed, which we denoted by p_i, $where\ 1 \leq i \leq 8$. In addition, setup time is required if job j will be processed immediately after job i on the same machine and denoted by s_{ij}, $where\ 1 \leq i, j \leq 8\ and\ i \neq j$. Table 3.1. gives the processing times of each job and Table 3.2. provides the setup time at unit processing speed.

Table 3.1. Processing Times for Example 1

Jobs (i)	**Job 1**	**Job 2**	**Job 3**	**Job 4**	**Job 5**	**Job 6**	**Job 7**	**Job 8**
Processing Time (p_i)	4	6	3	5	5	6	5	3

Table 3.2. Setup Time Matrix for Example 1

Setup Time (s_{ij})	Job 1	Job 2	Job 3	Job 4	Job 5	Job 6	Job 7	Job 8
Job 1	-	6	3	5	3	6	5	2
Job 2	6	-	1	2	2	4	5	5
Job 3	3	1	-	2	6	4	4	2
Job 4	5	2	2	-	2	1	2	1
Job 5	3	2	6	2	-	4	1	2
Job 6	6	4	4	1	4	-	2	1
Job 7	5	5	4	2	1	2	-	3
Job 8	2	5	2	1	2	1	3	-

Figure 3.1. and Figure 3.2. show two Gantt charts for production schedules in identical and uniform parallel machines with one setup operator . In the Gantt chart, numbers in white bars indicate job indices, and black bars represent setup operations. Numbers in the bottom denote time stamps; for example, in Figure 3.1., Machine 1 finishes at 20 while the completion time of Machines 2 is 18 and Machine 3 is 19. Assume that jobs are processed in their index order, the jobs processed for the first time on each machine do not require setups, and there is no dedicated machine constraint. When jobs are processed on identical parallel machines, their makespan is 20, as illustrated in Figure 3.1. However, the processing speed of machine k is denoted by v_k, $where\ 1 \leq k \leq 3$. When the machines have different speeds, i.e., v_1, v_2, and v_3 are 0.8, 1, and 1.2, respectively, the processing times change. Therefore, the processing time of job i on machine k is calculated with this formulation: $p_{ik} = p_i/v_k$. In addition, for a particular machine k, the setup time of job j immediately after job i is calculated by $s_{ijk} = s_{ij}/v_k$. After these calculations, the schedule becomes the same as the Gantt chart in Figure 3.2. The makespan is 22.5.

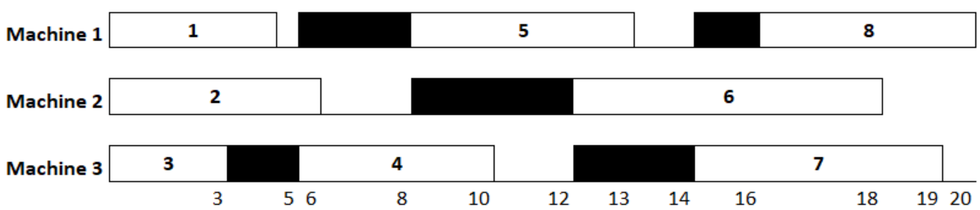

Figure 3.1. Schedule in Identical Parallel Machine for Example 1

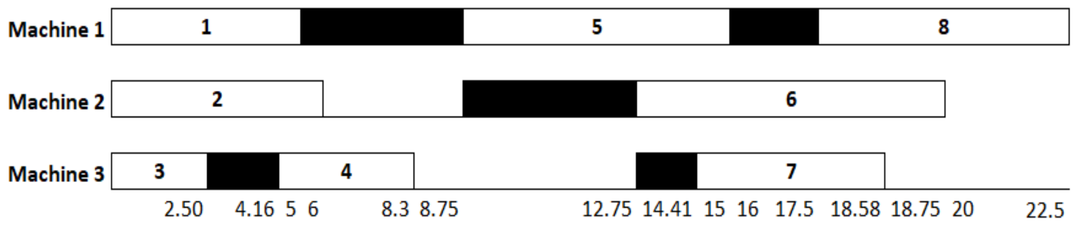

Figure 3.2. Schedule in Uniform Parallel Machine for Example 1

We now propose an integer programming (IP) model for uniform parallel machine scheduling with sequence-dependent setup times with the objective of minimizing the maximum of machine completion times (makespan), which we denoted $Q/ST_{sd}/C_{max}$. It is assumed that the machines are not malfunctioning and that jobs are available at time zero. The machines are ready at the beginning of the scheduling period. Data is deterministic and known in advance. Breakdown and maintenance times are not considered and machines are always available if not busy. All jobs must be completed without interruption. Each job should be completed on one machine and machines can perform only one job at a given time.

Most importantly, machines are uniform (Q), where the machines have different speeds but each machine processes at a consistent rate. Moreover, setup times are sequence-dependent which means the setup times not only depend on the job currently being scheduled but also on the immediate preceding job. Also, setup times depend on the machine speeds. An IP model is proposed for the problem. We list the rest of our sets, indices, parameters and decision variables.

Sets and indices

N: Set of jobs to be processed, $N = \{1,2, \dots, n\}$

M: Set of uniform parallel machines, $M = \{1,2, \dots, m\}$

i, j: Indices of jobs, where $i, j \in N$

k: Index of machines, where $k \in M$

Parameters

v_k : processing speed of machine k

p_i : processing time for job i at the base speed

p_{ik} : processing time for job i on machine k

s_{ij} : setup time of job j immediately after job i at the base speed

s_{ijk} : setup time of job j immediately after job i on machine k

L: a large number

Decision Variables

$$x_{ijk} = \begin{cases} 1, & \text{if job } i \text{ is immediately after job } j \text{ on machine } k \\ 0, & \text{otherwise} \end{cases}$$

$$y_{ik} = \begin{cases} 1, & \text{if job } i \text{ is assigned to machine } k \\ 0, & \text{otherwise} \end{cases}$$

C_i = completion time of job i

c_{max} = the maximum copletion time (makespan)

Next, we provide the IP formulation of the uniform machine scheduling problem with sequence-dependent setup times.

Model

$$Minimize\ C_{max} \tag{1}$$

$$Subject\ to$$

$$\sum_{i=1}^{n} x_{0ik} = 1 \qquad \forall k \tag{2}$$

$$C_i \geq s_{0ik} + p_{ik} - L(1 - x_{0ik}) \qquad \forall i, k \tag{3}$$

$$\sum_{i=0, i \neq j}^{n} x_{ijk} = y_{jk} \qquad \forall j, k \tag{4}$$

$$\sum_{j=1, i \neq j}^{n} x_{ijk} \leq y_{ik} \qquad \forall i, k \tag{5}$$

$$C_j \geq C_i + (s_{ijk} + p_{jk}) - L(1 - x_{ijk}) \qquad \forall i, j, k \tag{6}$$

$$\sum_{k=1}^{m} y_{ik} = 1 \qquad \forall i \tag{7}$$

$$C_{max} \geq C_{J,} \qquad \forall j \tag{8}$$

$$C_i \geq 0 \qquad \forall i \tag{9}$$

$$x_{ijk} \in \{0,1\} \qquad \forall \mathrm{i}, j, k \tag{10}$$

$$y_{ik} \in \{0,1\} \qquad \forall i, k \tag{11}$$

For notational convenience, we introduced a dummy job, $i = 0$, for each machine. The objective function (1) is the minimization of the maximum completion time of all jobs. Constraint set (2) ensures that the dummy job 0 is the initial job for each machine. Constraint (3) ensures that the completion of the very first job of every machine is at least as much as the sum of its setup time and the processing time at that machine.

Constraint set (4) establishes that the precedence relationship exists between jobs assigned to a particular machine. Similarly, if a real job is assigned to a machine, it can succeed by at most one job. The job in the last position of the sequence on a machine will not have a succeeding job. Constraint sets (4) and (5) together verify that n jobs are assigned to m machines. They also ensure that if job i immediately precedes job j on machine k, then both jobs i and j belong to machine k. Constraint set (6) guarantees that the finishing time of a real job in a sequence of a machine will be more than or equal to the sum of processing time of the current job, the sequence-dependent setup time and the finishing time of the preceding job. Constraint set (7) ensures that a real job is assigned to exactly one machine. The makespan is obtained in Constraint (8) and Constraint (9) ensure a non-negative completion time for regular jobs. Finally, Constraint (10) and (11) specify that the variables in the model are binary.

The problem we consider belongs to the set of NP-hard problems. Garey and Johnson (1979) mentioned that a problem is NP-hard if an algorithm for solving it can be translated into one for solving any NP (Nondeterministic Polynomial-Time) problem.

Owing to its academic and industrial importance, the UPMSP has been extensively investigated in recent decades and heuristics algorithms represent an alternative way of dealing with large-sized problems or combinatorial optimization problems. Despite the available technologies in today’s sector, the computation time for the exact methods for most large-sized problems in the literature is very long and unrealizable because the time to obtain the optimal solution to the NP-hard problem increases exponentially as the size of the problem increases.

The difficulty of getting the optimal solution in a reasonable time motivated many studies, including ours, to consider heuristic solutions to obtain a near-optimal solution in a reasonable time. As a result, exact methods become ineffective for large problems, and heuristics approaches can significantly reduce computation time without necessarily leading to the optimal solution. As we mentioned in the literature review, many heuristic models such as Genetic Algorithm (GA), Simulated Annealing (SA), Variable Neighborhood Search (VNS) have been studied for parallel machine scheduling problems. This thesis proposes a simple randomized heuristic with an improvement subroutine. The following sections give a detailed description of the developed heuristic.

Algorithm

Set $\mathcal{E}$, $C_{max} = \infty$, $\Delta C_{max} = \infty$, N

while $\Delta C_{max} > \mathcal{E}$

repeat N times

$C_{max}^{best} = \infty$

Assign jobs randomly to machines in a random order

Calculate C_m , $\forall_m$ and C_{max}^{new}

$C_{max}^{old} = \infty$

while $C_{max}^{new} < C_{max}^{old}$

Assign last job of machine with the largest C_m to the machine with the smallest C_m

$C_{max}^{old} = C_{max}^{new}$

Calculate C_{max}^{new} and C_m , $\forall_m$ with the updated schedule

If $C_{max}^{old} < C_{max}^{best}$ $C_{max}^{best} = C_{max}^{old}$

$$\Delta C_{max} = 100 \times \frac{C_{max} - C_{max}^{best}}{C_{max}}$$

$C_{max} = C_{max}^{best}$

end

Figure 3.3. Pseudo-Code of the Randomized Heuristic

The proposed heuristic is based on iterating a two-stage algorithm as long as we obtain a schedule with a smaller objective function value. In the first stage of this two-stage algorithm, we randomly assign jobs to the machines in random order. To elaborate, we choose one of the jobs from the set of jobs via a roulette wheel selection and randomly assign that job to one of the machines as the last assigned job that machine. We update the set of jobs by removing the assigned job from the set. We continue this assignment until the set of jobs is empty. At the end of this first algorithm stage, we obtain a feasible solution. We calculate the job completion times and the makespan for this schedule and proceed to the second stage with this information. The second stage of the algorithm relies on the observation that in the optimal solution, completion times of the last jobs of every machine are in proximity of each other. Hence, we identify the job with the largest completion time, remove that job from its assigned machine's list and append it to the job list of the machine with the shortest completion time as the last job. When we make this modification in the schedule, we also take machine-specific processing times of the jobs. After modifying the schedule, we update all the metrics, i.e., job completion times and the makespan, and repeat this improvement process until makespan does not improve. We iteratively run this two-stage algorithm by new random assignments at every iteration until the objective function value does not improve. We present the pseudo-code of this heuristic in Figure 3.3.

Heuristic Example. Suppose that there are fifteen jobs and three machines in the manufacturing system. Table 3.3. shows the processing times for each job and machine index. Table 3.4. gives the setup with a matrix. Processing times are randomly generated between 1 and 6 and setup times are generated between 1 and 5. Also, jobs are randomly assigned to machines.

Table 3.3. Processing Times and Job Index for Heuristic Example

Jobs	**Job 1**	**Job 2**	**Job 3**	**Job 4**	**Job 5**	**Job 6**	**Job 7**	**Job 8**	**Job 9**	**Job 10**	**Job 11**	**Job 12**	**Job 13**	**Job 14**	**Job 15**
Processing Times	3	3	4	1	2	3	2	3	3	4	5	2	6	5	2
Machine Index	2	2	2	2	3	1	3	2	2	3	3	1	1	3	2

Table 3.4. Setup Time Matrix for Heuristic Example

Jobs	**Job 1**	**Job 2**	**Job 3**	**Job 4**	**Job 5**	**Job 6**	**Job 7**	**Job 8**	**Job 9**	**Job 10**	**Job 11**	**Job 12**	**Job 13**	**Job 14**	**Job 15**
Job 1	0	3	4	2	1	4	1	2	1	4	4	2	3	3	5
Job 2	3	0	4	4	4	1	3	4	3	3	2	2	4	3	4
Job 3	4	4	0	1	3	2	2	1	4	4	4	4	1	2	5
Job 4	2	4	1	0	1	3	1	4	1	1	4	2	1	3	3
Job 5	1	4	3	1	0	4	4	4	3	3	2	3	3	2	5
Job 6	4	1	2	3	4	0	3	3	4	3	4	2	4	1	2
Job 7	1	3	2	1	4	3	0	2	3	2	3	1	4	4	1
Job 8	2	4	1	4	4	3	2	0	1	3	1	3	4	1	1
Job 9	1	3	4	1	3	4	3	1	0	4	1	2	2	3	4
Job 10	4	3	4	1	3	3	2	3	4	0	2	1	4	1	1
Job 11	4	2	4	4	2	4	3	1	1	2	0	2	1	1	4
Job 12	2	2	4	2	3	2	1	3	2	1	2	0	3	4	3
Job 13	3	4	1	1	3	4	4	4	2	4	1	3	0	1	2
Job 14	3	3	2	3	2	1	4	1	3	1	1	4	1	0	1
Job 15	5	4	5	3	5	2	1	1	4	1	4	3	2	1	0

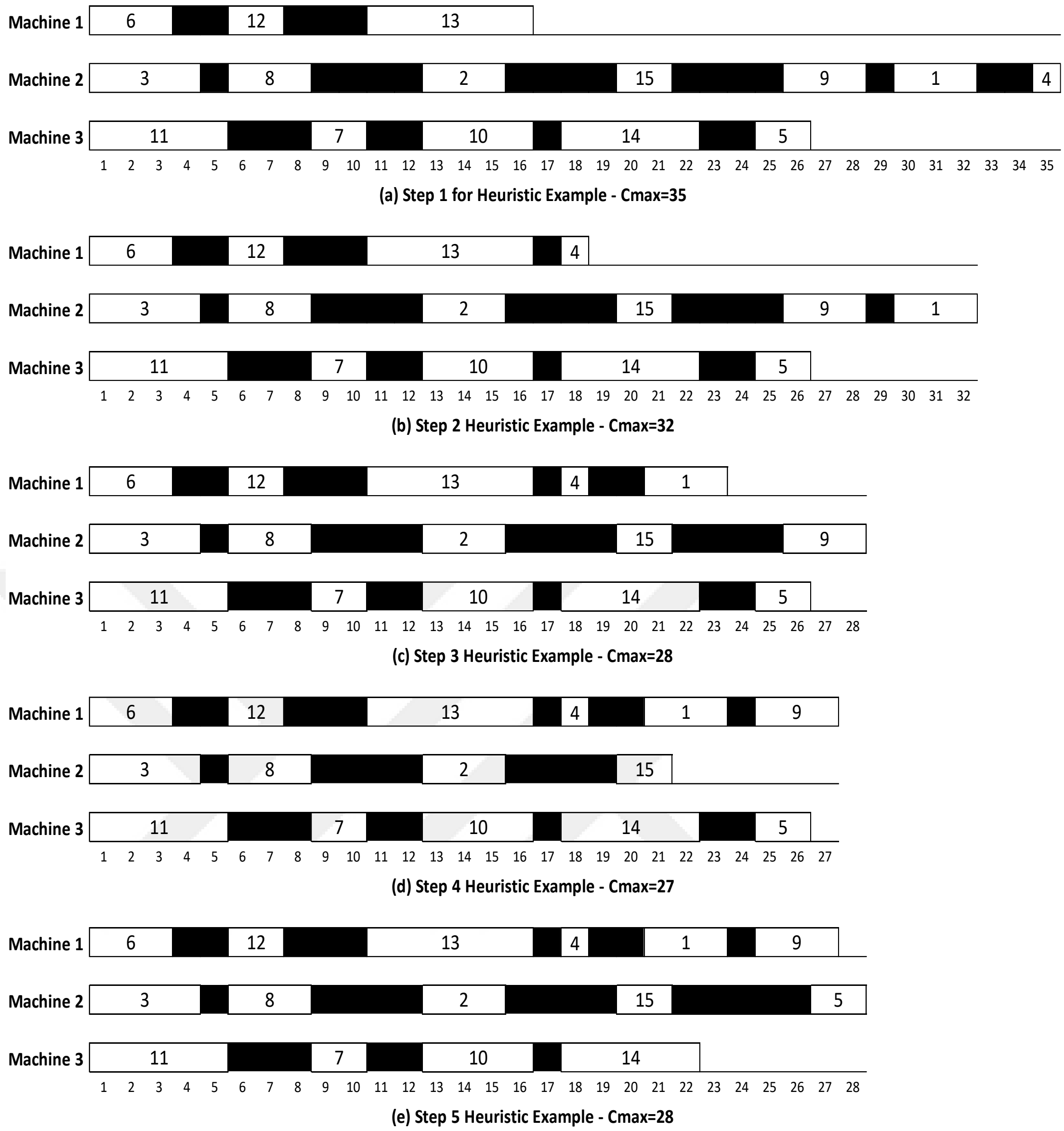

Figure 3.4. Gannt Charts for Heuristic Example

According to the Table 3.3., we can see from the machine indices without any order yet, the list of jobs to be processed on the machines is as follows: $\text{Machine 1 Job List} = \{6,12,13\}$, $\text{Machine 2 Job List} = \{1,2,3,4,8,9,15\}$, and $\text{Machine 3 Job List} = \{5,7,10,11,14\}$. Note that numbers in white bars indicate job indices, and black bars represent set up operations. Numbers in the bottom denote timestamps in the Gannt charts. The first step in our heuristic modeling, jobs assigned to machines are processed in random order. To elaborate, we choose one of the jobs from the set of jobs and randomly assign that job to one of the machines as the last assigned job that machine. We update the set of jobs by removing the assigned job from the set. We continue this assignment until the set of jobs is empty. For example, jobs assigned to machine 3 are 11,7,10,14,5, respectively. The important thing is that each job transition has setup time. At this point, it should be noted that setup time is required for each job transition. Then the completion time is calculated for each machine. You can see all the steps in Figure 3.4. In the first step (a), completion times are 16, 26 and 35 for the machines and makespan is 35. In the improvement step, we remove the last job of the machine with the longest completion time from that machine and assign it to the machine with the shortest finishing time as the last job. When we make this modification in the schedule, we also take the jobs' processing times and setup times. For instance, in step 3 (c), the longest completion time is on the second machine. So, we cut Job 9, the last job of the second machine, and assigned it to the first machine with the shortest completion time. Thus, the makespan reduced from 28 to 27. One can see the improved schedule in step 4 (d). After modifying the schedule, we update all the metrics, i.e., job completion times and the makespan, and repeat this improvement process until makespan does not improve. In step 5 (e), there is no improvement in makespan. So, we need to stop the algorithm.

CHAPTER 4
COMPUTATIONAL STUDY

This section is devoted to analyzing the experiments carried out to minimize the maximum completion time in a parallel machines scheduling problem with sequence-dependent setup times. To test the performance of our proposed heuristic, we have conducted a computational study. In this computation study, we have generated 320 test instances. We solved the IP model of the problem for each test instance until we obtained the optimal solution or 1-hour runtime was exceeded, whichever occurs first. We compared heuristic solutions with the IP model solutions. We solved the IP model with CPLEX 12.8 (Zarnikau, 1994) on a computer Intel Core i5-1035G1 CPU 1.19 GHz. Next, we will provide the details of our testbed and comparison results.

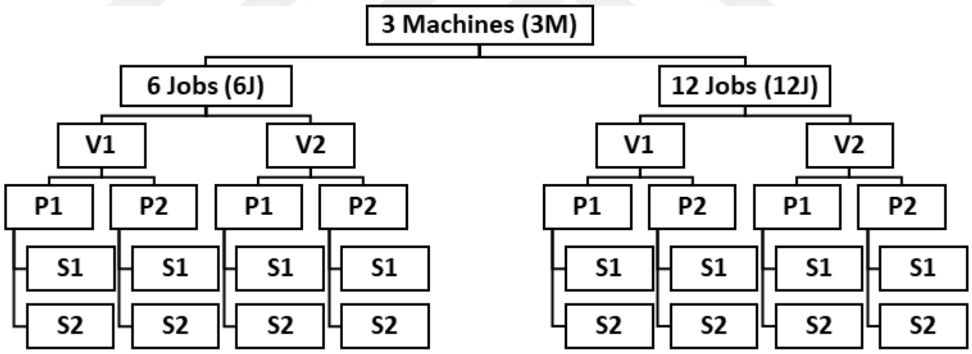

Figure 4.1. Combinations for 3 Machines

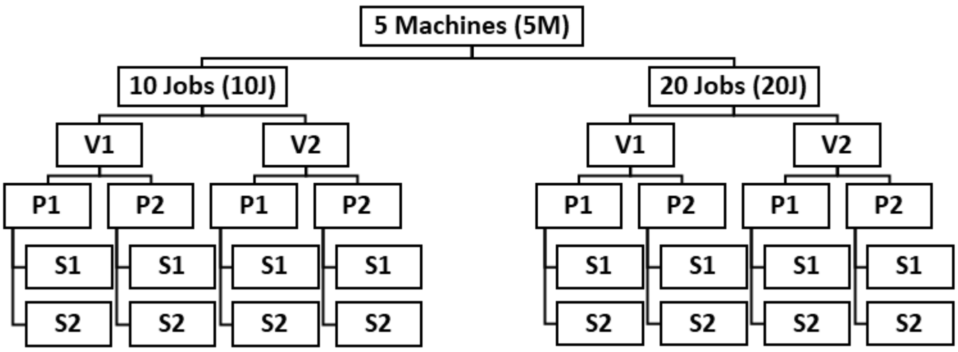

Figure 4.2. Combinations for 5 Machines

Our testbed comprises four different (number of machines-number of jobs) pairs. In two of these combinations, the number of jobs is twice the number of machines; and in the other combination, the number of jobs is four times the number of machines. The varieties we use are: (3M-6J), (3M-12J), (5M-10J), (5M-20J). We have constructed a setting for each pair for machine speeds, processing times, and setup times. The full factorial design is the most commonly utilized procedure for two or more factors. Our Design of Experiment (DOE) consists of all possible combinations of levels for all factors. . Kolisch et al. (1999) considered that the total number of experiments for studying *k* factors at 2-levels is 2^k. In our problem, we used 3 factors at 2-levels. These factors are machine speeds (V1, V2), process times (P1, P2) and setup times (S1, S2). Therefore, we have $8 \times 4 = 32$ combinations.

As shown in Figure 4.1 and Figure 4.2, a total of 32 combinations were created in our experimental study, 16 combinations for 3 machines and 16 combinations for 5 machines. For example, when considering 3 machines (3M) combinations, we must first look at how many jobs we select. For this, we have two options, 6 jobs or 12 jobs. When we continue with 6 jobs (6J), we will have to decide on the machine speed. Assuming that we have chosen the machine speed as the first machine speed range (V1), we must decide on the processing time in the next step. We have two options for this. Let's assume that we continue with the second processing time interval (P2); the situation we need to decide in the last step will be the setup time interval. Again, we have two options for this. Assuming we choose the first range (S1), finally, we will have the combination of 3M-6J-V1-P2-S1 as the encoding.

Table 4.1. Testbed for the Computational Study

Factors			Notation
3 Machines - 6 Jobs			3M-6J
3 Machines - 12 Jobs			3M-12J
5 Machines - 10 Jobs			5M-10J
5 Machines - 20 Jobs			5M-20J
Machine Speed Range 1	=	U [0.50,1.50]	V1
Machine Speed Range 2	=	U [0.85,1.15]	V2
Process Times Range 1	=	U [5,45]	P1
Process Times Range 2	=	U [15,35]	P2
Setup Times Range 1	=	U [1,5]	S1
Setup Times Range 2	=	U [2,4]	S2

Machine speeds are randomly generated either from Uniform [0.50,1.50] distribution or Uniform [0.85,1.15]. Processing times are generated either from Uniform [5,45] distribution or Uniform [15,35], and setup times are generated either from Uniform [1,5] distribution or Uniform [2,4]. We also have ten different variates for each of 32 combinations. We summarize our testbed in Table 4.1.

Table 4.2. Average CPLEX and Heuristic Solutions

#	Combinations	CPLEX			Heuristic		% Gap Between Cmax
		Optimal	**Avg. C_{max}**	**Avg. Run Time (seconds)**	**Avg. C_{max}**	**Avg. Run Time (seconds)**	
1	3M-6J-V1-P1-S1	1	53.819	0.128	54.7181	0.037	1.67%
2	3M-6J-V1-P1-S2	1	54.451	0.156	55.1682	0.0235	1.32%
3	3M-6J-V1-P2-S1	1	53.958	0.164	54.7086	0.0324	1.39%
4	3M-6J-V1-P2-S2	1	54.635	0.147	55.1078	0.047	0.87%
5	3M-6J-V2-P1-S1	1	51.559	0.133	52.2999	0.0346	1.44%
6	3M-6J-V2-P1-S2	1	51.979	0.158	52.4425	0.0333	0.89%
7	3M-6J-V2-P2-S1	1	51.964	0.147	52.379	0.0289	0.80%
8	3M-6J-V2-P2-S2	1	52.254	0.167	52.4746	0.0278	0.42%
9	3M-12J-V1-P1-S1	0	100.236	3600.494	104.878	0.0476	4.63%
10	3M-12J-V1-P1-S2	0	102.260	3600.599	105.339	0.0474	3.01%
11	3M-12J-V1-P2-S1	0	103.110	3600.616	107.831	0.0655	4.58%
12	3M-12J-V1-PS-S2	0	105.152	3600.653	107.947	0.1011	2.66%
13	3M-12J-V2-P1-S1	0	99.544	3600.518	103.875	0.0595	4.35%
14	3M-12J-V2-P1-S2	0	101.515	3600.670	104.242	0.066	2.69%
15	3M-12J-V2-P2-S1	0	102.431	3601.070	107.159	0.0579	4.62%
16	3M-12J-V2-P2-S2	0	104.211	3600.924	107.281	0.0882	2.95%
17	5M-10J-V1-P1-S1	1	53.515	34.908	56.6446	0.0541	5.85%
18	5M-10J-V1-P1-S2	1	53.671	42.700	56.8202	0.0721	5.87%
19	5M-10J-V1-P2-S1	1	56.562	66.163	60.8473	0.0656	7.58%
20	5M-10J-V1-P2-S2	1	57.333	80.002	61.0528	0.0907	6.49%
21	5M-10J-V2-P1-S1	1	52.207	38.833	55.5779	0.1111	6.46%
22	5M-10J-V2-P1-S2	1	52.536	46.024	55.5269	0.0908	5.69%
23	5M-10J-V2-P2-S1	1	53.344	26.022	56.1497	0.0948	5.26%
24	5M-10J-V2-P2-S2	1	53.784	30.281	56.5074	0.1553	5.06%
25	5M-20J-V1-P1-S1	0	103.580	3600.728	108.895	0.1688	5.13%
26	5M-20J-V1-P1-S2	0	105.403	3600.780	109.938	0.0667	4.30%
27	5M-20J-V1-P2-S1	0	108.010	3601.233	114.281	0.1128	5.81%
28	5M-20J-V1-PS-S2	0	109.874	3600.917	114.465	0.0838	4.18%
29	5M-20J-V2-P1-S1	0	104.247	3600.644	110.179	0.0661	5.69%
30	5M-20J-V2-P1-S2	0	105.790	3600.724	111.048	0.0643	4.97%
31	5M-20J-V2-P2-S1	0	105.580	3600.431	112.523	0.0991	6.58%
32	5M-20J-V2-P2-S2	0	107.593	3600.403	113.1	0.1045	5.12%

The computational results of the proposed heuristic algorithm are presented in this section. To evaluate the performance of the proposed heuristic methods, 320 instances with varying job sizes and machine sizes are developed. All proposed heuristic algorithms are coded in Python programming language and heuristic algorithms solve all instances within a few seconds on an Intel Core i5 2.40 GHz computer. The relative percent deviations (RPD) from the optimal results are reported for each heuristic algorithm as below:

$$RHD\ (\%\ Gap) = \frac{C_{max}^{heuristic} - C_{max}^{optimal}}{C_{max}^{optimal}} \times 100 \qquad (1)$$

It was explained in the previous sections that an experimental design was created by producing ten different instances for each of the 32 data combinations. In Table 4.2, solutions were obtained by taking the average of each combination's ten different instance files. If the algorithm could find the optimal value within the 1-hour time limit, the third column in the table was indicated as 1. If it could not reach the optimal within this time constraint, it was specified as 0. Also, the % Gap results in the eighth column were obtained using Equation (1) explained above.

Table 4.3. CPLEX - Average Solutions

	V1	V2	P1	P2	S1	S2
3M-6J	54.22	51.94	52.95	53.20	52.83	53.33
3M-12J	102.69	101.93	100.89	103.73	101.33	103.28
5M-10J	55.27	52.97	52.98	55.26	53.91	54.33
5M-20J	106.72	105.80	104.75	107.76	105.35	107.16

Table 4.4. Heuristic - Average Solutions

	V1	V2	P1	P2	S1	S2
3M-6J	54.93	52.40	53.66	53.67	53.53	53.80
3M-12J	106.50	105.64	104.58	107.55	105.94	106.20
5M-10J	58.84	55.94	56.14	58.64	57.30	57.48
5M-20J	111.89	111.71	110.01	113.59	111.47	112.14

The most important result of this study is that while CPLEX 12.8 takes 22.88 minutes on average and the heuristic algorithm achieves these results only in 0.062 minutes. Moreover, Tables 4.3 and 4.4 show the average solutions of the CPLEX model and heuristic model. The average solutions obtained with the heuristic have an approximately 4% Gap value for an average CPLEX solution.

Table 4.5. Mean of Heuristic % Gap

	V1	V2	P1	P2	S1	S2
3M-6J	1.33%	0.97%	1.47%	0.83%	1.34%	0.96%
3M-12J	3.74%	3.68%	3.68%	3.74%	4.59%	2.83%
5M-10J	6.43%	5.75%	6.16%	6.02%	6.35%	5.83%
5M-20J	4.90%	5.60%	5.10%	5.40%	5.85%	4.65%

Table 4.6. Median of Heuristic % Gap

	V1	V2	P1	P2	S1	S2
3M-6J	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
3M-12J	0.88%	0.00%	0.00%	2.03%	1.47%	0.88%
5M-10J	0.78%	0.00%	0.74%	0.00%	0.00%	0.35%
5M-20J	0.39%	1.21%	0.39%	0.87%	1.20%	0.39%

In Tables 4.5 and 4.6 you can see the impact of all our factors on the results. Table 4.5. indicates the mean calculations of these % Gap values, while Table 4.6. is for the median calculations.

Table 4.7. % Gap Deviation for All Instances

	≤1%	1%-5%	5%-10%	≥10%
Optimal	50	56	48	6
Not Optimal	1	102	57	0
%	15.94%	49.38%	32.81%	1.88%

We compared the results we obtained after solving the mathematical modeling and the proposed heuristic algorithm, and we calculated the % Gap deviations at certain percentage intervals using Equation (1). The percentage values in the last row of Table 4.7. compare the solutions in that range with 320 instances. For example, 48 optimal and 57 non-optimal results were found between 5% and 10% gap intervals. In other words, a total of 105 instances gave a solution in this range, and this has a rate of 32.81% among 320 instances.

As we explained in the heuristic section, the system we have established consists of two main parts: random assignment and improvement subroutine. The contribution of the improvement subroutine step to the overall performance of the heuristic is 73.34%

on average. This means that the improvements made on the initial solution created due to random assignment have made the system much more intelligent.

Another significant result in this experimental design is when the CPLEX 12.8 runs with a given one-hour time limitation. 160 out of 320 instances can be found optimal result and proposed randomized heuristic results found 19 out of these 160 instances. This means that our randomized heuristic algorithm can achieve optimal results with a rate of 11.88% on average.

CHAPTER 5
CONCLUSION AND FUTURE RESEARCH

This thesis considered a uniform parallel machine scheduling problem with sequence-dependent setup times to minimize the maximum completion time (makespan). We present an IP formulation, which clearly describes our problem and can be used to obtain optimal solutions for small-sized problems. Our problem is NP-hard which means that the time to obtain the optimal solution to the problem increases exponentially as the size of the problem increases. Therefore, we propose a randomized heuristic algorithm with two stages: random assignment and improvement subroutine. In the first step of this two-stage algorithm, we assign random jobs to the machines in random order and continue this assignment until the job list is empty. We get a feasible solution at the end of this first algorithm step. Then, we calculate the completion times and the makespan and move on to the second step with this information. In the second step, we first determine the last job of the machine with the largest completion time and remove it from the machine with the shortest completion time as the last job. When we do these modifications, we consider the machine speeds and repeat this improvement process until makespan does not improve. The primary purpose of this improvement subroutine step is balancing loads of the machines, reducing the completion times, and reducing the makespan.

Our thesis has four different (number of machines-number of jobs) pairs for data. In two of these combinations, the number of jobs is twice the number of machines and in the other combination, the number of jobs is four times the number of machines. The performance of the algorithm is tested with 320 instance sets. We create these instances using the full factorial design of experiments (DOE). The total number of experiments for studying k factors at 2-levels is 2^k. In our problem, we used 3 factors at 2-levels. These factors are machine speeds, process times and setup times. Also, all these data are randomly generated either from specified Uniform intervals. Therefore, we have 32 combinations and all combinations have ten replications.

In this study, we compared the results after solving the mathematical modeling and the proposed heuristic algorithm, and we calculated the % Gap deviations at certain percentage intervals using. This study's most important numerical result is that the proposed mathematical model takes 22.88 minutes on average, and the heuristic algorithm achieves these results only in 0.062 minutes. The heuristic has an approximately 4% Gap value for an average. Also, the contribution of the improvement subroutine step to the overall performance of the heuristic is 73.34% on average. Another significant result in this experimental design is when the CPLEX runs with a given one-hour time limitation, we can find the optimal solution in half of the instances in CPLEX, whereas this number is only 19 out of 160 instances. This means that our randomized heuristic algorithm can achieve optimal results with a rate of 11.88% on average. Finally, we can conclude that the proposed randomized heuristic algorithm is very efficient and provides highly acceptable solutions.

Future work can be addressed as follows: (1) The heuristic algorithm presented in this study can be more effective and intelligent. (2) While generating the data, the size of the problem can be increased by expanding the uniform intervals, or benchmarking examples in the literature can be used for computational experimentation.

REFERENCES

Abreu, L. R., & Prata, B. A. (2019). A genetic algorithm with neighborhood search procedures for unrelated parallel machine scheduling problem with sequence-dependent setup times. *Journal of Modelling in Management, 15(3)*, 809-828. doi: 10.1108/JM2-12-2018-0209

Afzalirad, M., & Rezaeian, J. (2016, May 12). Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions. *Computers & Industrial Engineering, 98*, 40-52. doi:10.1016/j.cie.2016.05.020

Ahmarofi, A. A., Abidin, N. Z., & Ramli, R. (2017). Effect of manpower factor on semiautomatic production line completion time: A system dynamics approach. *Journal of Mechanical Engineering and Sciences, 11(2)*, 2567-2580. doi:10.15282/jmes.11.2.2017.1.0235

Al-Harkan, I. M., & Qamhan, A. A. (2019, November 26). Optimize unrelated parallel machines scheduling problems with multiple limited additional resources, sequence-dependent setup times and release date constraints. *IEEE Access, 7*, 171533 - 171547. doi:10.1109/ACCESS.2019.2955975

Allahverdi, A. (2015). The third comprehensive survey on scheduling problems with setup times/costs. *European Journal of Operational Research, 246*, 345-378.

Allahverdi, A., & Soroush, H. (2008). The significance of reducing setup times/setup costs. *European Journal of Operational Research, 187*, 978–984.

Allahverdi, A., Gupta, J. N., & Aldowaisan, T. (1999). A review of scheduling research involving setup considerations. *Omega, International Journal of Management Science, 27*, 219-239.

Allahverdi, A., Ng, C. T., Cheng, T., & Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research, 187*, 985–1032.

Angel-Bello, F., Avalos-Rosales, O., Álvarez, A., & Cardona-Valdés, Y. (2018, July 5). Including preventive maintenance activities in an unrelated parallel machine environment with dependent setup times. *Computers & Industrial Engineering, 123*, 364–377. doi:10.1016/j.cie.2018.07.006

Arbaoui, T., & Yalaoui, F. (2016). An exact approach for the identical parallel machine scheduling problem with sequence-dependent setup times and the job splitting property. *International Conference on Industrial Engineering and Engineering Management (IEEM).* Bali, Indonesia: IEEE.

Arbaoui, T., & Yalaoui, F. (2018). Solving the unrelated parallel machine scheduling problem with additional resources using constraint programming. *Intelligent Information and Database Systems: 10th Asian Conference, ACIIDS* (s. 716-

725). ong Hoi City, Vietnam: Springer. doi:10.1007/978-3-319-75420-8_67

Armentano, V. A., & Franca Filho, M. F. (2007). Minimizing total tardiness in parallel machine scheduling with setup times: An adaptive memory-based GRASP approach. *European Journal of Operational Research, 183*, 100–114. doi:10.1016/j.ejor.2006.09.077

Avalos-Rosales, O., & Angel-Bello, F. (2015). Efficient metaheuristic algorithm and re-formulations for the unrelated parallel machine scheduling problem with sequence and machine-dependent setup times. *International Journal of Advanced Manufacturing Technology*, *76*, 1705–1718. doi:10.1007/s00170-014-6390-6

Báez, S., Angel-Bello, F., Alvarez, A., & Melián-Batista, B. (2019). A hybrid metaheuristic algorithm for a parallel machine scheduling problem with dependent setup times. *Computers & Industrial Engineering*, *131*, 295–305. doi:10.1016/j.cie.2019.03.051

Balakrishnan, N., Kanet, J. J., & Sridharan, V. (1999). Early/tardy scheduling with sequence dependent setups on uniform parallel machines. *Computers & Operations Research, 26*, 127-141.

Beezão, A. C., Cordeau, J.-F., & Laporte, G. (2017). Scheduling identical parallel machines with tooling constraints. *European Journal of Operational Research, 257*, 834–844. doi:10.1016/j.ejor.2016.08.008

Behnamian, J. (2015). *Survey on fuzzy shop scheduling.* New York, NY, USA: Springer.

Behnamian, J., Zandieh, M., & Fatemi Ghomi, S. (2009). Parallel-machine scheduling problems with sequence-dependent setup times using an ACO, SA and VNS hybrid algorithm. *Expert Systems with Applications, 36,* 9637–9644. doi:10.1016/j.eswa.2008.10.007

Behnamian, J., Zandieh, M., & Ghomi, S. F. (2011). Bi-objective parallel machines scheduling with sequence-dependent setup times using hybrid metaheuristics and weighted min–max technique. *Soft Computing, 15*, 1313–1331.

Bektur, G., & Sarac, T. (2019). A mathematical model and heuristic algorithms for an unrelated parallel machine scheduling problem with sequence-dependent setup times, machine eligibility restrictions and a common server. *Computers and Operations Research, 103*, 46-63. doi:10.1016/j.cor.2018.10.010

Bosman, T., Frascaria, D., Olver, N., Sitters, R., & Stougie, L. (2019). Fixed-order scheduling on parallel machines. *Conference 20th International Conference on Integer Programming and Combinatorial Optimization, IPCO* (s. 88-100). United States: Ann Arbor, Springer Verlag.

Caniyilmaz, E., Benli, B., & Ilkay, M. S. (2015). An artificial bee colony algorithm approach for unrelated parallel machine scheduling with processing set restrictions, job sequence-dependent setup times, and due date. *International Journal of Advanced Manufacturing Technology, 77*, 2105–2115. doi:10.1007/s00170-014-6614-9

Chen, C. L., & Chen, C. L. (2009, August 26). Hybrid metaheuristics for unrelated parallel machine scheduling with sequence-dependent setup times. *International Journal of Advanced Manufacturing Technology, 43*, 161–169. doi:10.1007/s00170-008-1692-1

Chen, J. F. (2009, February 19). Scheduling on unrelated parallel machines with sequence- and machine-dependent setup times and due-date constraints. *International Journal of Advanced Manufacturing Technology, 44*, 1204–1212. doi:10.1007/s00170-008-1917-3

Cheng, C. Y., Pourhejazy, P., Ying, K. C., Li, S. F., & Chang, A. C. (2020). Learning-based metaheuristic for scheduling unrelated parallel machines with uncertain setup times. *IEEE Access, 8*, 74065-74082.

Cheng, T., & Sin, C. (1990). A state-of-the-art review of parallel-machine scheduling research. *European Journal of Operational Research*, *47*, 271-292.

Cota, L. P., Guimarães, F. G., Ribeiro, R. G., Meneghini, I. R., Oliveira, F. B., Souza, M. J., & Siarry, P. (2019, October 31). An adaptive multi-objective algorithm based on decomposition and large neighborhood search for a green machine scheduling problem. *Swarm and Evolutionary Computation, 51*, 100601. doi:10.1016/j.swevo.2019.100601

Diana, R. O., França Filho, M. F., Souza, S. R., & Almeida Vitor, F. (2015). An immune-inspired algorithm for an unrelated parallel machines'scheduling problem with sequence and machine dependent setup-times for makespan minimisation. *Neurocomputing, 163*, 94–105.

Driessel, R., & Mönch, L. (2009). Scheduling jobs on parallel machines with sequence-dependent setup times, precedence constraints, and ready times using variable neighborhood search. *International Conference on Computers & Industrial Engineering.* Troyes, France: IEEE.

Driessel, R., & Mönch, L. (2011). Variable neighborhood search approaches for scheduling jobs on parallel machines with sequence-dependent setup times, precedence constraints, and ready times. *Computers & Industrial Engineering, 61*, 336–345.

Ekici, A., Elyasi, M., Ozener, O. O., & Sarıkaya, M. B. (2019). An application of unrelated parallel machine scheduling with sequence-dependent setups at Vestel Electronics. *Computers and Operations Research, 111*, 130–140. doi:10.1016/j.cor.2019.06.007

Expósito-Izquierdo, C., Angel-Bello, F., Melián-Batista, B., Alvarez, A., & Báez, S. (2019). A metaheuristic algorithm and simulation to study the effect of learning or tiredness on sequence-dependent setup times in a parallel machine scheduling problem. *Expert Systems With Applications, 117*, 62–74. doi:10.1016/j.eswa.2018.09.041

Ezugwu, A. E. (2019). Enhanced symbiotic organisms search algorithm for unrelated parallel machines manufacturing scheduling with setup times. *Knowledge-Based Systems, 172*, 15–32.

Ezugwu, A. E., & Akutsah, A. F. (2018). An improved firefly algorithm for the unrelated parallel machines scheduling problem with sequence-dependent setup times. *IEEE Access, 6*, 54459-54478.

Ezugwu, A. E., Adeleke, O. J., & Viriri, S. (2018). Symbiotic organisms search algorithm for the unrelated parallel machines scheduling with sequence-dependent setup times. PloS One, 13(7), e0200030. [Online]., doi:10.1371/journal.pone.0200030

Fanjul-Peyro, L., Perea, F., & Ruiz, R. (2017). Models and matheuristics for the unrelated parallel machine scheduling problem with additional resources. *European Journal of Operational Research, 260*, 482–493.

Fanjul-Peyro, L., Ruiz, R., & Perea, F. (2019). Reformulations and an exact algorithm for unrelated parallel machine scheduling problems with setup times. *Computers and Operations Research, 101*, 173–182. doi:10.1016/j.cor.2018.07.007

Garey, M. R., & Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* San Francisco: Freeman.

Gedik, R., Kalathia, D., Egilmez, G., & Kirac, E. (2018). A constraint programming approach for solving unrelated parallel machine scheduling problem. *Computers & Industrial Engineering, 121*, 139–149. doi:10.1016/j.cie.2018.05.014

Gedik, R., Rainwater, C., Nachtmann, H., & Pohl, E. A. (2016). Analysis of a parallel machine scheduling problem with sequence dependent setup times and job availability intervals. *European Journal of Operational Research, 251*, 640–650.

Graham, R. L., Lawler, E. L., Lenstra, J. K., & Rinnooy Kan, A. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics, 5*, 287-326.

Hamzadayi, A., & Yildiz, G. (2007). Modeling and solving static m identical parallel machines scheduling problem with a common server and sequence dependent setup times. *Computers & Industrial Engineering, 106*, 287–298.

Hsu, H. M., Hsiung, Y., Chen, Y. Z., & Wu, M. C. (2009). A GA methodology for the scheduling of yarn-dyed textile production. *Expert Systems with Applications, 36*, 12095–12103.

Hu, H., Ng, K. K., & Qin, Y. (2016). Robust parallel machine scheduling problem with uncertainties and sequence-dependent setup time. *Hindawi Publishing Corporation Scientific Programming*, *2016*, 1-13. doi:10.1155/2016/5127253

Joo, C. M., & Kim, B. S. (2012). Parallel machine scheduling problem with ready times, due times and sequence-dependent setup times using meta-heuristic algorithms. *Engineering Optimization, 44*, 1021–1034.

Kim, J.-G., Song, S., & Jeong, B. (2020). Minimising total tardiness for the identical parallel machine scheduling problem with splitting jobs and sequence-dependent setup times. *International Journal of Production Research, 58*, 1628–1643.

Kolisch, R., Schwindt, C., & Sprecher, A. (1999). Benchmark instances for project scheduling problems. *Project Scheduling*, 197-212. doi:10.1007/978-1-4615-5533-9_9

Kongsri, P., & Buddhakulsomsiri, J. (2020). A mixed integer programming model for unrelated parallel machine scheduling problem with sequence dependent setup time to minimize makespan and total tardiness. *7th International Conference on Industrial Engineering and Applications.* Bangkok, Thailand: IEEE. doi:10.1109/ICIEA49774.2020.9102086

Kopanos, G. M., Lainez, J. M., & Puigjaner, L. (2009). An efficient mixed-integer linear programming scheduling framework for addressing sequence-dependent setup issues in batch plants. *Industrial Engineering Chemical Research, 48,* 6346–6357.

Lee, Z.-J., Lin, S.-W., & Ying, K.-C. (2010). Scheduling jobs on dynamic parallel machines with sequence-dependent setup times. *International Journal of Advanced Manufacturing Technology, 47*, 773–781.

Lin, S. W., & Ying, K. C. (2014). ABC-based manufacturing scheduling for unrelated parallel machines with machine-dependent and job sequence-dependent setup times. *Computers & Operations Research, 51*, 172–181.

Lopes, M. J., & Carvalho, J. (2007). A branch-and-price algorithm for scheduling parallel machines with sequence dependent setup times. *European Journal of Operational Research*, *176*, 1508–1527. doi:10.1016/j.ejor.2005.11.001

Loveland, J. L., Monkman, S. K., & Morrice, D. J. (2007). Dell uses a new production-scheduling algorithm to accommodate increased product variety. *Interfaces, 37*, 209–219.

McNaughton R., (1959). Scheduling with deadlines and loss functions. *Management Science*, *6*, 1-12.

Müller, F. M., Araújo, O. B., Stefanello, F., & Zanetti, M. (2015, January). MIP–Based neighborhood search for the unrelated parallel machine scheduling problem with sequence and machine-dependent setup times. *Revista de Administração* *da UFSM, 7*, 506-523. doi:10.5902/1983465913305

Naderi-Beni, M., Ghobadian, E., Ebrahimnejad, S., & Tavakkoli-Moghaddam, R. (2014, March 31). Fuzzy bi-objective formulation for a parallel machine scheduling problem with machine eligibility restrictions and sequence-dependent setup times. *International Journal of Production Research, 52*, 5799–5822,. doi:10.1080/00207543.2014.916430

Oyetunji, E. O. (2009). Some common performance measures in scheduling problems: Review article. *Research Journal of Applied Sciences, Engineering and Technology, 2*, 6-9.

Ozer, E. A., & Sarac, T. (2019). MIPmodels and amatheuristic algorithm for an identical parallel machine scheduling problem under multiple copies of shared resources constraints. *TOP, 27*, 94–124.

Park, T., Lee, T., & Kim, C. O. (2012). Due-date scheduling on parallel machines with job splitting and sequence-dependent major/minor setup times. *International Journal of Advanced Manufacturing Technology, 59*, 325–333.

Paula, M. R., Mateus, G. R., & Ravetti, M. G. (2010). A non-delayed relax-and-cut algorithm for scheduling problems with parallel machines, due dates and sequence-dependent setup times. *Computers & Operations Research, 37*, 938--949. doi:10.1016/j.cor.2009.07.006

Pinedo, M. L. (2012). *Scheduling Theory, Algorithms, and Systems.* New York, NY, USA: Springer.

Queiroz, T. A., & Mundim, L. R. (2019). Multiobjective pseudo-variable neighborhood descent for a bicriteria parallel machine scheduling problem with setup time. *International Transactions in Operation Research*, 1-23.

Rauchecker, G., & Schryen, G. (2019, April). Using high performance computing for unrelated parallel machine scheduling with sequence-dependent setup times: Development and computational evaluation of a parallel branch-and-price algorithm. *Computers and Operations Research, 104*, 338–357. doi:10.1016/j.cor.2018.12.020

Rocha, P. L., Ravetti, M. G., Mateus, G. R., & Pardalos, P. M. (2008). Exact algorithms for a scheduling problem with unrelated parallel machines and sequence and machine-dependent setup times. *Computers & Operations Research, 35*, 1250 – 1264. doi:10.1016/j.cor.2006.07.015

Safaei, S., Naderi, R., Sohrabi, A., & Hatami, A. (2015, December). Scheduling of unrelated parallel machines using two multi objective genetic algorithms with sequence-dependent setup times and precedent constraints. *International Journal of Advanced Design and Manufacturing Technology, 8(4)*, 63-74.

Silva, L. G., Rego, M. F., Assis, L. P., & Andrade, A. V. (2019). Algorithms for the unrelated parallel machine scheduling problem with sequence dependent setup times. *37th International Conference of the Chilean Computer Science Society*. Santiago, Chile: IEEE.

Soares a, L. C., & Carvalho, M. A. (2020). Biased random-key genetic algorithm for scheduling identical parallel machines with tooling constraints. *European Journal of Operational Research, 285*, 955–964.

Tahar, D. N., Yalaoui, F., Chu, C., & Amodeo, L. (2006). A linear programming approach for identical parallel machine scheduling with job splitting and sequence-dependent setup times. *International Journal of Production Economics, 99*, 63–73.

Tavakkoli-Moghaddam, R., Taheri, F., Bazzazi, M., Izadi, M., & Sassani, F. (2009, March 3). Design of a genetic algorithm for bi-objective unrelated parallel machines scheduling with sequence-dependent setup times and precedence constraints. *Computers & Operations Research, 36*, 3224--3230. doi:10.1016/j.cor.2009.02.012

Tozzo, E., Cotrim, S. L., Galdamez, E. V., & Leal, G. C. (2018). A genetic algorithm and variable neighborhood search for the unrelated parallel machine scheduling problem with sequence dependent setup time. *Acta Scientiarum Technology, 40(1)*. e36607. [Online]. doi: 10.4025/actascitechnol.v40i1.36607

Turker, K. A., & Sel, C. (2011). Scheduling two parallel machines with sequence-dependent setups and A single server. *Gazi University Journal of Science, 24*, 113-123.

Vallada, E., & Ruiz, R. (2011). A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times. *European Journal of Operational Research, 211*, 612–622. doi:10.1016/j.ejor.2011.01.011

Wang, L., Wang, S., & Zheng, X. (2016). A hybrid estimation of distribution algorithm for unrelated parallel machine scheduling with sequence-dependent setup times. *IEEE/CAA Journal of Automatica Sinica, 3*, 235-246.

Xu, X., Cui, W., Lin, J., & Qian, Y. (2013). Robust makespan minimisation in identical parallel machine scheduling problem with interval data. *International Journal of Production Research, 51*, 3532–3548.

Ying, K. C. (2012). Scheduling identical wafer sorting parallel machines with sequence-dependent setup times using an iterated greedy heuristic. *International Journal of Production Research, 50*, 2710–2719.

Ying, K. C., & Liao, C. J. (2004). An ant colony system for permutation & fow-shop sequencing. *Computers & Operations Research, 31*, 791–801.

Ying, K.-C., & Cheng, H.-M. (2010). Dynamic parallel machine scheduling with sequence-dependent setup times using an iterated greedy heuristic. *Expert Systems with Applications, 37*, 2848–2852.

Zeidi, J. R., & Hosseini, S. M. (2015, May 24). Scheduling unrelated parallel machines with sequence-dependent setup times. *International Journal of Advanced Manufacturing Technology, 81*, 1487–1496. doi:10.1007/s00170-015-7215-y

Zeidi, R. J., Zarei, M., & Shokoufi, K. (2017, December). Pareto-based multi-criteria evolutionary algorithm for a parallel machines scheduling problem with sequence-dependent setup times. *International Journal of Engineering, Transactions, 30(12)*, 1863-1869. doi:10.5829/ije.2017.30.12c.07

APPENDIX 1 – CPLEX SOLUTIONS FOR ALL INSTANCES

		CPLEX Solution				Lower Bound	
		Opt?	C_{max}	Run Time (sec.)	% Gap	C_{max}	% Gap
GROUP 1	3M-6J-V1-P1-S1-D1	1	45.256	0.109	0%	42.405	6.30%
	3M-6J-V1-P1-S1-D2	1	52.264	0.172	0%	47.511	9.09%
	3M-6J-V1-P1-S1-D3	1	63.314	0.14	0%	58.535	7.55%
	3M-6J-V1-P1-S1-D4	1	67.331	0.156	0%	62.057	7.83%
	3M-6J-V1-P1-S1-D5	1	68.858	0.094	0%	65.972	4.19%
	3M-6J-V1-P1-S1-D6	1	40.941	0.094	0%	37.528	8.34%
	3M-6J-V1-P1-S1-D7	1	42.482	0.14	0%	40.497	4.67%
	3M-6J-V1-P1-S1-D8	1	44.025	0.141	0%	39.905	9.36%
	3M-6J-V1-P1-S1-D9	1	57.809	0.157	0%	54.594	5.56%
	3M-6J-V1-P1-S1-D10	1	55.908	0.078	0%	48.16	13.86%
GROUP 2	3M-6J-V1-P1-S2-D1	1	47.313	0.141	0%	42.405	10.37%
	3M-6J-V1-P1-S2-D2	1	51.89	0.141	0%	47.511	8.44%
	3M-6J-V1-P1-S2-D3	1	62.409	0.156	0%	58.535	6.21%
	3M-6J-V1-P1-S2-D4	1	67.331	0.078	0%	62.057	7.83%
	3M-6J-V1-P1-S2-D5	1	69.421	0.203	0%	65.972	4.97%
	3M-6J-V1-P1-S2-D6	1	42.305	0.188	0%	37.528	11.29%
	3M-6J-V1-P1-S2-D7	1	43.98	0.156	0%	40.497	7.92%
	3M-6J-V1-P1-S2-D8	1	45.359	0.188	0%	39.905	12.02%
	3M-6J-V1-P1-S2-D9	1	57.839	0.235	0%	54.594	5.61%
	3M-6J-V1-P1-S2-D10	1	56.664	0.078	0%	48.16	15.01%
GROUP 3	3M-6J-V1-P2-S1-D1	1	49.37	0.25	0%	44.975	8.90%
	3M-6J-V1-P2-S1-D2	1	63.865	0.157	0%	60.014	6.03%
	3M-6J-V1-P2-S1-D3	1	54.411	0.172	0%	52.523	3.47%
	3M-6J-V1-P2-S1-D4	1	69.563	0.14	0%	64.835	6.80%
	3M-6J-V1-P2-S1-D5	1	52.065	0.14	0%	48.732	6.40%
	3M-6J-V1-P2-S1-D6	1	46.399	0.141	0%	41.907	9.68%
	3M-6J-V1-P2-S1-D7	1	50.794	0.156	0%	46.361	8.73%
	3M-6J-V1-P2-S1-D8	1	48.164	0.125	0%	46.261	3.95%
	3M-6J-V1-P2-S1-D9	1	55.844	0.125	0%	52.508	5.97%
	3M-6J-V1-P2-S1-D10	1	49.108	0.234	0%	46.365	5.59%
GROUP 4	3M-6J-V1-P2-S2-D1	1	50.398	0.188	0%	44.975	10.76%
	3M-6J-V1-P2-S2-D2	1	65.33	0.172	0%	60.014	8.14%
	3M-6J-V1-P2-S2-D3	1	55.173	0.109	0%	52.523	4.80%
	3M-6J-V1-P2-S2-D4	1	71.174	0.156	0%	64.835	8.91%
	3M-6J-V1-P2-S2-D5	1	52.509	0.156	0%	48.732	7.19%
	3M-6J-V1-P2-S2-D6	1	46.019	0.125	0%	41.907	8.94%
	3M-6J-V1-P2-S2-D7	1	49.951	0.156	0%	46.361	7.19%
	3M-6J-V1-P2-S2-D8	1	50.84	0.172	0%	46.261	9.01%
	3M-6J-V1-P2-S2-D9	1	55.844	0.11	0%	52.508	5.97%
	3M-6J-V1-P2-S2-D10	1	49.108	0.125	0%	46.365	5.59%

		CPLEX Solution				Lower Bound	
		Opt?	C_{max}	Run Time (sec.)	% Gap	C_{max}	% Gap
GROUP 5	3M-6J-V2-P1-S1-D1	1	46.369	0.156	0%	43.509	6.17%
	3M-6J-V2-P1-S1-D2	1	38.307	0.125	0%	35.187	8.14%
	3M-6J-V2-P1-S1-D3	1	67.85	0.14	0%	60.693	10.55%
	3M-6J-V2-P1-S1-D4	1	53.589	0.109	0%	48.766	9.00%
	3M-6J-V2-P1-S1-D5	1	62.387	0.094	0%	59.17	5.16%
	3M-6J-V2-P1-S1-D6	1	41.24	0.172	0%	39.225	4.89%
	3M-6J-V2-P1-S1-D7	1	44.407	0.11	0%	41.309	6.98%
	3M-6J-V2-P1-S1-D8	1	40.878	0.156	0%	38.311	6.28%
	3M-6J-V2-P1-S1-D9	1	56.758	0.094	0%	52.992	6.64%
	3M-6J-V2-P1-S1-D10	1	63.805	0.172	0%	51.887	18.68%
GROUP 6	3M-6J-V2-P1-S2-D1	1	47.394	0.157	0%	43.509	8.20%
	3M-6J-V2-P1-S2-D2	1	39.979	0.219	0%	35.187	11.99%
	3M-6J-V2-P1-S2-D3	1	66.881	0.187	0%	60.693	9.25%
	3M-6J-V2-P1-S2-D4	1	53.589	0.187	0%	48.766	9.00%
	3M-6J-V2-P1-S2-D5	1	61.877	0.203	0%	59.17	4.37%
	3M-6J-V2-P1-S2-D6	1	42.887	0.094	0%	39.225	8.54%
	3M-6J-V2-P1-S2-D7	1	44.885	0.141	0%	41.309	7.97%
	3M-6J-V2-P1-S2-D8	1	41.748	0.109	0%	38.311	8.23%
	3M-6J-V2-P1-S2-D9	1	57.659	0.141	0%	52.992	8.09%
	3M-6J-V2-P1-S2-D10	1	62.894	0.141	0%	51.887	17.50%
GROUP 7	3M-6J-V2-P2-S1-D1	1	50.079	0.219	0%	46.146	7.85%
	3M-6J-V2-P2-S1-D2	1	48.623	0.141	0%	44.447	8.59%
	3M-6J-V2-P2-S1-D3	1	58.157	0.094	0%	54.459	6.36%
	3M-6J-V2-P2-S1-D4	1	54.196	0.172	0%	50.95	5.99%
	3M-6J-V2-P2-S1-D5	1	56.233	0.141	0%	52.74	6.21%
	3M-6J-V2-P2-S1-D6	1	45.627	0.156	0%	43.801	4.00%
	3M-6J-V2-P2-S1-D7	1	50.164	0.11	0%	46.342	7.62%
	3M-6J-V2-P2-S1-D8	1	46.967	0.125	0%	44.414	5.44%
	3M-6J-V2-P2-S1-D9	1	53.995	0.172	0%	50.966	5.61%
	3M-6J-V2-P2-S1-D10	1	55.602	0.141	0%	49.953	10.16%
GROUP 8	3M-6J-V2-P2-S2-D1	1	49.411	0.125	0%	46.146	6.61%
	3M-6J-V2-P2-S2-D2	1	48.623	0.172	0%	44.447	8.59%
	3M-6J-V2-P2-S2-D3	1	58.157	0.094	0%	54.459	6.36%
	3M-6J-V2-P2-S2-D4	1	55.182	0.281	0%	50.95	7.67%
	3M-6J-V2-P2-S2-D5	1	55.152	0.218	0%	52.74	4.37%
	3M-6J-V2-P2-S2-D6	1	46.786	0.234	0%	43.801	6.38%
	3M-6J-V2-P2-S2-D7	1	50.164	0.156	0%	46.342	7.62%
	3M-6J-V2-P2-S2-D8	1	47.969	0.109	0%	44.414	7.41%
	3M-6J-V2-P2-S2-D9	1	55.168	0.14	0%	50.966	7.62%
	3M-6J-V2-P2-S2-D10	1	55.931	0.14	0%	49.953	10.69%

		CPLEX Solution				Lower Bound	
		Opt?	C_{max}	Run Time (sec.)	% Gap	C_{max}	% Gap
GROUP 9	3M-12J-V1-P1-S1-D1	0	109.776	3600.36	52%	105.117	4.24%
	3M-12J-V1-P1-S1-D2	0	74.013	3600.453	41%	70.429	4.84%
	3M-12J-V1-P1-S1-D3	0	110.393	3600.86	48%	106.158	3.84%
	3M-12J-V1-P1-S1-D4	0	80.741	3600.281	47%	76.653	5.06%
	3M-12J-V1-P1-S1-D5	0	105.959	3600.391	48%	101.011	4.67%
	3M-12J-V1-P1-S1-D6	0	86.795	3600.297	47%	81.708	5.86%
	3M-12J-V1-P1-S1-D7	0	138.631	3600.281	45%	131.798	4.93%
	3M-12J-V1-P1-S1-D8	0	83.196	3600.953	46%	78.909	5.15%
	3M-12J-V1-P1-S1-D9	0	93.325	3600.485	45%	89.9	3.67%
	3M-12J-V1-P1-S1-D10	0	119.529	3600.578	48%	112.76	5.66%
GROUP 10	3M-12J-V1-P1-S2-D1	0	112.295	3600.547	52%	105.117	6.39%
	3M-12J-V1-P1-S2-D2	0	75.539	3600.484	48%	70.429	6.76%
	3M-12J-V1-P1-S2-D3	0	112.46	3600.844	49%	106.158	5.60%
	3M-12J-V1-P1-S2-D4	0	82.993	3600.625	49%	76.653	7.64%
	3M-12J-V1-P1-S2-D5	0	107.978	3600.312	48%	101.011	6.45%
	3M-12J-V1-P1-S2-D6	0	87.817	3600.344	44%	81.708	6.96%
	3M-12J-V1-P1-S2-D7	0	142.141	3600.266	46%	131.798	7.28%
	3M-12J-V1-P1-S2-D8	0	85.493	3600.656	49%	78.909	7.70%
	3M-12J-V1-P1-S2-D9	0	95.378	3600.344	48%	89.9	5.74%
	3M-12J-V1-P1-S2-D10	0	120.509	3601.563	51%	112.76	6.43%
GROUP 11	3M-12J-V1-P2-S1-D1	0	96.167	3600.594	53%	91.716	4.63%
	3M-12J-V1-P2-S1-D2	0	77.883	3600.703	51%	73.571	5.54%
	3M-12J-V1-P2-S1-D3	0	100.793	3600.625	47%	96.243	4.51%
	3M-12J-V1-P2-S1-D4	0	86.892	3600.422	52%	82.737	4.78%
	3M-12J-V1-P2-S1-D5	0	102.088	3600.656	55%	97.405	4.59%
	3M-12J-V1-P2-S1-D6	0	89.867	3600.578	52%	85.34	5.04%
	3M-12J-V1-P2-S1-D7	0	139.407	3601.109	54%	132.954	4.63%
	3M-12J-V1-P2-S1-D8	0	91.277	3600.563	46%	87.75	3.86%
	3M-12J-V1-P2-S1-D9	0	90.504	3600.531	51%	87.019	3.85%
	3M-12J-V1-P2-S1-D10	0	156.225	3600.375	57%	149.007	4.62%
GROUP 12	3M-12J-V1-P2-S2-D1	0	97.773	3600.218	54%	91.716	6.19%
	3M-12J-V1-P2-S2-D2	0	79.053	3600.531	51%	73.571	6.93%
	3M-12J-V1-P2-S2-D3	0	102.78	3600.688	51%	96.243	6.36%
	3M-12J-V1-P2-S2-D4	0	89.316	3600.531	54%	82.737	7.37%
	3M-12J-V1-P2-S2-D5	0	103.32	3600.641	54%	97.405	5.72%
	3M-12J-V1-P2-S2-D6	0	91.883	3600.609	54%	85.34	7.12%
	3M-12J-V1-P2-S2-D7	0	141.65	3600.781	55%	132.954	6.14%
	3M-12J-V1-P2-S2-D8	0	94.581	3600.687	51%	87.75	7.22%
	3M-12J-V1-P2-S2-D9	0	92.593	3601.062	53%	87.019	6.02%
	3M-12J-V1-P2-S2-D10	0	158.572	3600.782	57%	149.007	6.03%

		CPLEX Solution				Lower Bound	
		Opt?	C_{max}	Run Time (sec.)	% Gap	C_{max}	% Gap
GROUP 13	3M-12J-V2-P1-S1-D1	0	104.099	3601.032	51%	100.032	3.91%
	3M-12J-V2-P1-S1-D2	0	88.33	3601.078	50%	82.876	6.17%
	3M-12J-V2-P1-S1-D3	0	122.033	3600.666	50%	116.344	4.66%
	3M-12J-V2-P1-S1-D4	0	87.073	3600.082	51%	81.652	6.23%
	3M-12J-V2-P1-S1-D5	0	102.204	3600.076	51%	97.45	4.65%
	3M-12J-V2-P1-S1-D6	0	91.695	3600.081	55%	87.341	4.75%
	3M-12J-V2-P1-S1-D7	0	104.483	3601.07	49%	99.178	5.08%
	3M-12J-V2-P1-S1-D8	0	85.043	3600.076	54%	79.901	6.05%
	3M-12J-V2-P1-S1-D9	0	103.456	3601.012	49%	99.327	3.99%
	3M-12J-V2-P1-S1-D10	0	107.027	3600.005	49%	101.753	4.93%
GROUP 14	3M-12J-V2-P1-S2-D1	0	107.161	3600.656	53%	100.032	6.65%
	3M-12J-V2-P1-S2-D2	0	89.29	3600.765	48%	82.876	7.18%
	3M-12J-V2-P1-S2-D3	0	123.124	3600.656	54%	116.344	5.51%
	3M-12J-V2-P1-S2-D4	0	87.999	3600.391	47%	81.652	7.21%
	3M-12J-V2-P1-S2-D5	0	104.454	3600.906	47%	97.45	6.71%
	3M-12J-V2-P1-S2-D6	0	94.439	3601.281	50%	87.341	7.52%
	3M-12J-V2-P1-S2-D7	0	107.927	3600.212	54%	99.178	8.11%
	3M-12J-V2-P1-S2-D8	0	86.326	3600.068	57%	79.901	7.44%
	3M-12J-V2-P1-S2-D9	0	105.24	3601.009	51%	99.327	5.62%
	3M-12J-V2-P1-S2-D10	0	109.189	3600.759	50%	101.753	6.81%
GROUP 15	3M-12J-V2-P2-S1-D1	0	101.968	3600.687	48%	97.362	4.52%
	3M-12J-V2-P2-S1-D2	0	91.049	3600.969	49%	86.573	4.92%
	3M-12J-V2-P2-S1-D3	0	110.453	3600.766	49%	105.477	4.51%
	3M-12J-V2-P2-S1-D4	0	93.557	3600.843	51%	88.132	5.80%
	3M-12J-V2-P2-S1-D5	0	104.454	3601.125	50%	100.122	4.15%
	3M-12J-V2-P2-S1-D6	0	96.586	3600.922	50%	91.223	5.55%
	3M-12J-V2-P2-S1-D7	0	111.743	3600.656	51%	106.666	4.54%
	3M-12J-V2-P2-S1-D8	0	93.141	3600.75	51%	88.852	4.60%
	3M-12J-V2-P2-S1-D9	0	99.997	3603.609	48%	96.143	3.85%
	3M-12J-V2-P2-S1-D10	0	121.36	3600.375	52%	114.645	5.53%
GROUP 16	3M-12J-V2-P2-S2-D1	0	104.38	3600.688	52%	97.362	6.72%
	3M-12J-V2-P2-S2-D2	0	93.05	3600.969	48%	86.573	6.96%
	3M-12J-V2-P2-S2-D3	0	112.034	3602.578	51%	105.477	5.85%
	3M-12J-V2-P2-S2-D4	0	94.483	3600.672	50%	88.132	6.72%
	3M-12J-V2-P2-S2-D5	0	106.255	3600.984	49%	100.122	5.77%
	3M-12J-V2-P2-S2-D6	0	98.113	3601.029	50%	91.223	7.02%
	3M-12J-V2-P2-S2-D7	0	114.183	3601.15	48%	106.666	6.58%
	3M-12J-V2-P2-S2-D8	0	95.341	3600.043	49%	88.852	6.81%
	3M-12J-V2-P2-S2-D9	0	102.827	3601.056	49%	96.143	6.50%
	3M-12J-V2-P2-S2-D10	0	121.444	3600.071	52%	114.645	5.60%

		CPLEX Solution				Lower Bound	
		Opt?	C_{max}	Run Time (sec.)	% Gap	C_{max}	% Gap
GROUP 17	5M-10J-V1-P1-S1-D1	1	55.568	27.187	0%	53.084	4.47%
	5M-10J-V1-P1-S1-D2	1	41.421	42.219	0%	38.877	6.14%
	5M-10J-V1-P1-S1-D3	1	64.218	55.438	0%	60.465	5.84%
	5M-10J-V1-P1-S1-D4	1	41.156	27.594	0%	38.509	6.43%
	5M-10J-V1-P1-S1-D5	1	55.633	36.172	0%	50.31	9.57%
	5M-10J-V1-P1-S1-D6	1	56.83	33.391	0%	54.309	4.44%
	5M-10J-V1-P1-S1-D7	1	52.655	14.047	0%	48.979	6.98%
	5M-10J-V1-P1-S1-D8	1	44.221	35.281	0%	40.559	8.28%
	5M-10J-V1-P1-S1-D9	1	58.209	25.938	0%	55.213	5.15%
	5M-10J-V1-P1-S1-D10	1	65.235	51.812	0%	61.421	5.85%
GROUP 18	5M-10J-V1-P1-S2-D1	1	55.748	33.75	0%	53.084	4.78%
	5M-10J-V1-P1-S2-D2	1	42.328	57.844	0%	38.877	8.15%
	5M-10J-V1-P1-S2-D3	1	63.883	76.469	0%	60.465	5.35%
	5M-10J-V1-P1-S2-D4	1	42.281	32.875	0%	38.509	8.92%
	5M-10J-V1-P1-S2-D5	1	54.715	29.078	0%	50.31	8.05%
	5M-10J-V1-P1-S2-D6	1	57.643	29.203	0%	54.309	5.78%
	5M-10J-V1-P1-S2-D7	1	52.655	14.516	0%	48.979	6.98%
	5M-10J-V1-P1-S2-D8	1	43.789	48.75	0%	40.559	7.38%
	5M-10J-V1-P1-S2-D9	1	58.538	31.719	0%	55.213	5.68%
	5M-10J-V1-P1-S2-D10	1	65.128	72.797	0%	61.421	5.69%
GROUP 19	5M-10J-V1-P2-S1-D1	1	58.393	79.641	0%	55.776	4.48%
	5M-10J-V1-P2-S1-D2	1	43.174	54.844	0%	40.998	5.04%
	5M-10J-V1-P2-S1-D3	1	57.887	49.766	0%	53.768	7.12%
	5M-10J-V1-P2-S1-D4	1	46.307	39.218	0%	44.01	4.96%
	5M-10J-V1-P2-S1-D5	1	58.935	81.797	0%	55.724	5.45%
	5M-10J-V1-P2-S1-D6	1	57.643	42.11	0%	54.991	4.60%
	5M-10J-V1-P2-S1-D7	1	73.903	69.265	0%	68.766	6.95%
	5M-10J-V1-P2-S1-D8	1	50.252	67.672	0%	46.233	8.00%
	5M-10J-V1-P2-S1-D9	1	56.418	126.657	0%	53.508	5.16%
	5M-10J-V1-P2-S1-D10	1	62.703	50.656	0%	58.475	6.74%
GROUP 20	5M-10J-V1-P2-S2-D1	1	58.936	79.875	0%	55.776	5.36%
	5M-10J-V1-P2-S2-D2	1	44.021	71.641	0%	40.998	6.87%
	5M-10J-V1-P2-S2-D3	1	58.791	86.25	0%	53.768	8.54%
	5M-10J-V1-P2-S2-D4	1	47.043	40.594	0%	44.01	6.45%
	5M-10J-V1-P2-S2-D5	1	59.247	98.734	0%	55.724	5.95%
	5M-10J-V1-P2-S2-D6	1	58.654	38.094	0%	54.991	6.25%
	5M-10J-V1-P2-S2-D7	1	75.457	80.844	0%	68.766	8.87%
	5M-10J-V1-P2-S2-D8	1	50.252	66.625	0%	46.233	8.00%
	5M-10J-V1-P2-S2-D9	1	57.314	133.656	0%	53.508	6.64%
	5M-10J-V1-P2-S2-D10	1	63.614	103.703	0%	58.475	8.08%

		CPLEX Solution				Lower Bound	
		Opt?	C_{max}	Run Time (sec.)	% Gap	C_{max}	% Gap
GROUP 21	5M-10J-V2-P1-S1-D1	1	53.581	22.641	0%	50.45	5.84%
	5M-10J-V2-P1-S1-D2	1	46.472	34.172	0%	42.327	8.92%
	5M-10J-V2-P1-S1-D3	1	69.478	49.781	0%	62.354	10.25%
	5M-10J-V2-P1-S1-D4	1	40.401	15.828	0%	38.01	5.92%
	5M-10J-V2-P1-S1-D5	1	49.966	30.328	0%	46.471	6.99%
	5M-10J-V2-P1-S1-D6	1	52.174	34.672	0%	49.583	4.97%
	5M-10J-V2-P1-S1-D7	1	47.712	29.625	0%	44.484	6.77%
	5M-10J-V2-P1-S1-D8	1	42.203	26.235	0%	39.168	7.19%
	5M-10J-V2-P1-S1-D9	1	55.072	21.219	0%	52.779	4.16%
	5M-10J-V2-P1-S1-D10	1	65.006	123.828	0%	56.182	13.57%
GROUP 22	5M-10J-V2-P1-S2-D1	1	54.659	30.891	0%	50.45	7.70%
	5M-10J-V2-P1-S2-D2	1	46.09	22.547	0%	42.327	8.16%
	5M-10J-V2-P1-S2-D3	1	69.789	53.828	0%	62.354	10.65%
	5M-10J-V2-P1-S2-D4	1	40.751	26.656	0%	38.01	6.73%
	5M-10J-V2-P1-S2-D5	1	50.426	45.016	0%	46.471	7.84%
	5M-10J-V2-P1-S2-D6	1	52.78	40.688	0%	49.583	6.06%
	5M-10J-V2-P1-S2-D7	1	47.712	61.703	0%	44.484	6.77%
	5M-10J-V2-P1-S2-D8	1	42.063	30.625	0%	39.168	6.88%
	5M-10J-V2-P1-S2-D9	1	56.086	51.437	0%	52.779	5.90%
	5M-10J-V2-P1-S2-D10	1	65.006	96.844	0%	56.182	13.57%
GROUP 23	5M-10J-V2-P2-S1-D1	1	55.798	25.032	0%	53.224	4.61%
	5M-10J-V2-P2-S1-D2	1	48.369	38.75	0%	44.636	7.72%
	5M-10J-V2-P2-S1-D3	1	58.157	15.906	0%	55.448	4.66%
	5M-10J-V2-P2-S1-D4	1	45.721	32.11	0%	43.44	4.99%
	5M-10J-V2-P2-S1-D5	1	56.563	18.578	0%	53.933	4.65%
	5M-10J-V2-P2-S1-D6	1	52.864	20.266	0%	50.205	5.03%
	5M-10J-V2-P2-S1-D7	1	58.868	46.14	0%	54.126	8.06%
	5M-10J-V2-P2-S1-D8	1	47.106	15.437	0%	44.647	5.22%
	5M-10J-V2-P2-S1-D9	1	53.595	24.375	0%	51.149	4.56%
	5M-10J-V2-P2-S1-D10	1	56.403	23.625	0%	53.487	5.17%
GROUP 24	5M-10J-V2-P2-S2-D1	1	56.326	23.828	0%	53.224	5.51%
	5M-10J-V2-P2-S2-D2	1	47.658	43.688	0%	44.636	6.34%
	5M-10J-V2-P2-S2-D3	1	58.4	16.968	0%	55.448	5.05%
	5M-10J-V2-P2-S2-D4	1	46.967	29.937	0%	43.44	7.51%
	5M-10J-V2-P2-S2-D5	1	57.248	20.453	0%	53.933	5.79%
	5M-10J-V2-P2-S2-D6	1	53.069	21.562	0%	50.205	5.40%
	5M-10J-V2-P2-S2-D7	1	59.704	44.938	0%	54.126	9.34%
	5M-10J-V2-P2-S2-D8	1	47.958	49.031	0%	44.647	6.90%
	5M-10J-V2-P2-S2-D9	1	54.106	27.125	0%	51.149	5.47%
	5M-10J-V2-P2-S2-D10	1	56.403	25.281	0%	53.487	5.17%

		CPLEX Solution				Lower Bound	
		Opt?	C_{max}	Run Time (sec.)	% Gap	C_{max}	% Gap
GROUP 25	5M-20J-V1-P1-S1-D1	0	122.334	3602.328	76%	115.476	5.61%
	5M-20J-V1-P1-S1-D2	0	91.433	3600.469	73%	86.983	4.87%
	5M-20J-V1-P1-S1-D3	0	122.481	3600.391	78%	115.773	5.48%
	5M-20J-V1-P1-S1-D4	0	101.281	3600.531	73%	93.495	7.69%
	5M-20J-V1-P1-S1-D5	0	98.262	3600.422	67%	92.189	6.18%
	5M-20J-V1-P1-S1-D6	0	88.571	3600.89	67%	83.611	5.60%
	5M-20J-V1-P1-S1-D7	0	96.786	3600.484	75%	90.566	6.43%
	5M-20J-V1-P1-S1-D8	0	89.752	3600.344	67%	82.502	8.08%
	5M-20J-V1-P1-S1-D9	0	127.121	3600.421	76%	121.275	4.60%
	5M-20J-V1-P1-S1-D10	0	97.782	3601	69%	91.825	6.09%
GROUP 26	5M-20J-V1-P1-S2-D1	0	127.228	3601.156	85%	115.476	9.24%
	5M-20J-V1-P1-S2-D2	0	94.283	3600.344	76%	86.983	7.74%
	5M-20J-V1-P1-S2-D3	0	122.96	3600.953	73%	115.773	5.84%
	5M-20J-V1-P1-S2-D4	0	101.281	3600.609	79%	93.495	7.69%
	5M-20J-V1-P1-S2-D5	0	100.639	3600.438	74%	92.189	8.40%
	5M-20J-V1-P1-S2-D6	0	90.987	3600.968	76%	83.611	8.11%
	5M-20J-V1-P1-S2-D7	0	98.169	3600.375	70%	90.566	7.74%
	5M-20J-V1-P1-S2-D8	0	89.331	3601.219	85%	82.502	7.64%
	5M-20J-V1-P1-S2-D9	0	130.633	3601.204	80%	121.275	7.16%
	5M-20J-V1-P1-S2-D10	0	98.517	3600.532	80%	91.825	6.79%
GROUP 27	5M-20J-V1-P2-S1-D1	0	99.898	3600.609	72%	94.039	5.86%
	5M-20J-V1-P2-S1-D2	0	88.816	3600.421	73%	83.853	5.59%
	5M-20J-V1-P2-S1-D3	0	112.201	3604.672	79%	104.765	6.63%
	5M-20J-V1-P2-S1-D4	0	105.227	3601.985	73%	99.325	5.61%
	5M-20J-V1-P2-S1-D5	0	117.719	3601.094	75%	109.562	6.93%
	5M-20J-V1-P2-S1-D6	0	89.431	3600.406	71%	84.479	5.54%
	5M-20J-V1-P2-S1-D7	0	93.753	3600.75	69%	87.748	6.41%
	5M-20J-V1-P2-S1-D8	0	137.567	3600.5	79%	128.282	6.75%
	5M-20J-V1-P2-S1-D9	0	122.76	3600.453	80%	115.639	5.80%
	5M-20J-V1-P2-S1-D10	0	112.73	3601.437	79%	106.386	5.63%
GROUP 28	5M-20J-V1-P2-S2-D1	0	101.511	3601.546	72%	94.039	7.36%
	5M-20J-V1-P2-S2-D2	0	92.591	3600.75	74%	83.853	9.44%
	5M-20J-V1-P2-S2-D3	0	111.494	3600.578	78%	104.765	6.04%
	5M-20J-V1-P2-S2-D4	0	110.341	3601.782	74%	99.325	9.98%
	5M-20J-V1-P2-S2-D5	0	118.594	3601.031	75%	109.562	7.62%
	5M-20J-V1-P2-S2-D6	0	90.869	3600.89	71%	84.479	7.03%
	5M-20J-V1-P2-S2-D7	0	96.406	3600.797	70%	87.748	8.98%
	5M-20J-V1-P2-S2-D8	0	137.567	3600.531	79%	128.282	6.75%
	5M-20J-V1-P2-S2-D9	0	123.227	3600.515	80%	115.639	6.16%
	5M-20J-V1-P2-S2-D10	0	116.144	3600.75	77%	106.386	8.40%

		CPLEX Solution				Lower Bound	
		Opt?	C_{max}	Run Time (sec.)	% Gap	C_{max}	% Gap
GROUP 29	5M-20J-V2-P1-S1-D1	0	106.852	3600.312	81%	101.581	4.93%
	5M-20J-V2-P1-S1-D2	0	104.381	3600.484	69%	99.229	4.94%
	5M-20J-V2-P1-S1-D3	0	126.23	3600.625	69%	120.063	4.89%
	5M-20J-V2-P1-S1-D4	0	94.774	3600.406	74%	88.559	6.56%
	5M-20J-V2-P1-S1-D5	0	99.488	3600.437	69%	93.588	5.93%
	5M-20J-V2-P1-S1-D6	0	98.2	3600.469	63%	92.171	6.14%
	5M-20J-V2-P1-S1-D7	0	93.786	3601.25	81%	87.032	7.20%
	5M-20J-V2-P1-S1-D8	0	97.429	3601.61	72%	90.742	6.86%
	5M-20J-V2-P1-S1-D9	0	116.61	3600.328	67%	111.367	4.50%
	5M-20J-V2-P1-S1-D10	0	104.719	3600.516	76%	95.658	8.65%
GROUP 30	5M-20J-V2-P1-S2-D1	0	109.723	3600.375	76%	101.581	7.42%
	5M-20J-V2-P1-S2-D2	0	107.08	3600.735	75%	99.229	7.33%
	5M-20J-V2-P1-S2-D3	0	127.828	3600.375	77%	120.063	6.07%
	5M-20J-V2-P1-S2-D4	0	96.909	3600.328	78%	88.559	8.62%
	5M-20J-V2-P1-S2-D5	0	100.879	3600.297	71%	93.588	7.23%
	5M-20J-V2-P1-S2-D6	0	99.708	3600.562	79%	92.171	7.56%
	5M-20J-V2-P1-S2-D7	0	94.131	3601.782	61%	87.032	7.54%
	5M-20J-V2-P1-S2-D8	0	98.08	3600.454	67%	90.742	7.48%
	5M-20J-V2-P1-S2-D9	0	119.771	3600.032	68%	111.367	7.02%
	5M-20J-V2-P1-S2-D10	0	103.786	3602.297	90%	95.658	7.83%
GROUP 31	5M-20J-V2-P2-S1-D1	0	103.587	3600.359	75%	98.268	5.13%
	5M-20J-V2-P2-S1-D2	0	100.924	3600.468	69%	95.658	5.22%
	5M-20J-V2-P2-S1-D3	0	113.807	3600.36	74%	108.647	4.53%
	5M-20J-V2-P2-S1-D4	0	99.103	3600.453	71%	94.081	5.07%
	5M-20J-V2-P2-S1-D5	0	111.249	3600.297	77%	104.739	5.85%
	5M-20J-V2-P2-S1-D6	0	99.114	3600.515	70%	93.127	6.04%
	5M-20J-V2-P2-S1-D7	0	96.521	3600.375	78%	90.818	5.91%
	5M-20J-V2-P2-S1-D8	0	113.477	3600.515	77%	106.439	6.20%
	5M-20J-V2-P2-S1-D9	0	110.349	3600.453	74%	106.192	3.77%
	5M-20J-V2-P2-S1-D10	0	107.67	3600.516	74%	101.252	5.96%
GROUP 32	5M-20J-V2-P2-S2-D1	0	105.02	3600.312	75%	98.268	6.43%
	5M-20J-V2-P2-S2-D2	0	102.581	3600.5	77%	95.658	6.75%
	5M-20J-V2-P2-S2-D3	0	116.172	3600.312	74%	108.647	6.48%
	5M-20J-V2-P2-S2-D4	0	103.766	3600.391	77%	94.081	9.33%
	5M-20J-V2-P2-S2-D5	0	111.473	3600.469	76%	104.739	6.04%
	5M-20J-V2-P2-S2-D6	0	100.107	3600.313	72%	93.127	6.97%
	5M-20J-V2-P2-S2-D7	0	99.054	3600.453	73%	90.818	8.31%
	5M-20J-V2-P2-S2-D8	0	114.579	3600.391	79%	106.439	7.10%
	5M-20J-V2-P2-S2-D9	0	114.448	3600.453	77%	106.192	7.21%
	5M-20J-V2-P2-S2-D10	0	108.726	3600.438	72%	101.252	6.87%

APPENDIX 2 – IMPROVED HEURISTIC SOLUTIONS FOR ALL INSTANCES WITH REPITATION TIMES

		H 100 Cmax	H 200 Cmax	H 400 Cmax	H 800 Cmax	H 1600 Cmax	H-Best Cmax
GROUP 1	3M-6J-V1-P1-S1-D1	46.611	46.611				46.611
	3M-6J-V1-P1-S1-D2	53.221	52.265	52.265			52.265
	3M-6J-V1-P1-S1-D3	65.123	64.764				64.764
	3M-6J-V1-P1-S1-D4	67.658	67.331				67.331
	3M-6J-V1-P1-S1-D5	72.091	72.091				72.091
	3M-6J-V1-P1-S1-D6	40.941	40.941				40.941
	3M-6J-V1-P1-S1-D7	44.33	44.33				44.330
	3M-6J-V1-P1-S1-D8	46.139	44.693	44.693			44.693
	3M-6J-V1-P1-S1-D9	57.809	57.809				57.809
	3M-6J-V1-P1-S1-D10	58.066	56.346	56.346			56.346
GROUP 2	3M-6J-V1-P1-S2-D1	48.342	48.342				48.342
	3M-6J-V1-P1-S2-D2	51.89	51.89				51.890
	3M-6J-V1-P1-S2-D3	62.41	62.41				62.410
	3M-6J-V1-P1-S2-D4	67.331	67.331				67.331
	3M-6J-V1-P1-S2-D5	69.751	69.751				69.751
	3M-6J-V1-P1-S2-D6	44.353	44.353				44.353
	3M-6J-V1-P1-S2-D7	45.254	45.254				45.254
	3M-6J-V1-P1-S2-D8	47.848	47.848				47.848
	3M-6J-V1-P1-S2-D9	57.839	57.839				57.839
	3M-6J-V1-P1-S2-D10	56.664	56.664				56.664
GROUP 3	3M-6J-V1-P2-S1-D1	52.341	49.37	49.37			49.370
	3M-6J-V1-P2-S1-D2	65.991	65.991				65.991
	3M-6J-V1-P2-S1-D3	55.438	55.438				55.438
	3M-6J-V1-P2-S1-D4	72.422	72.422				72.422
	3M-6J-V1-P2-S1-D5	52.066	52.066				52.066
	3M-6J-V1-P2-S1-D6	46.94	46.47	46.47			46.470
	3M-6J-V1-P2-S1-D7	50.795	50.795				50.795
	3M-6J-V1-P2-S1-D8	52.178	48.695	48.695			48.695
	3M-6J-V1-P2-S1-D9	57.263	56.346	56.346			56.346
	3M-6J-V1-P2-S1-D10	49.568	49.493				49.493
GROUP 4	3M-6J-V1-P2-S2-D1	51.427	51.427				51.427
	3M-6J-V1-P2-S2-D2	69.064	65.331	65.331			65.331
	3M-6J-V1-P2-S2-D3	55.174	55.174				55.174
	3M-6J-V1-P2-S2-D4	72.422	72.422				72.422
	3M-6J-V1-P2-S2-D5	54.077	54.077				54.077
	3M-6J-V1-P2-S2-D6	46.4	46.4				46.400
	3M-6J-V1-P2-S2-D7	49.952	49.952				49.952
	3M-6J-V1-P2-S2-D8	50.84	50.84				50.840
	3M-6J-V1-P2-S2-D9	56.346	56.346				56.346
	3M-6J-V1-P2-S2-D10	49.109	49.109				49.109

*H is the reputation time

		H 100 C_{max}	H 200 C_{max}	H 400 C_{max}	H 800 C_{max}	H 1600 C_{max}	H-Best C_{max}
GROUP 5	3M-6J-V2-P1-S1-D1	46.686	46.686				46.686
	3M-6J-V2-P1-S1-D2	41.825	40.664	40.664			40.664
	3M-6J-V2-P1-S1-D3	67.851	67.851				67.851
	3M-6J-V2-P1-S1-D4	53.59	53.59				53.590
	3M-6J-V2-P1-S1-D5	65.966	62.388	62.388			62.388
	3M-6J-V2-P1-S1-D6	42.995	42.995				42.995
	3M-6J-V2-P1-S1-D7	44.407	44.407				44.407
	3M-6J-V2-P1-S1-D8	42.197	41.749	41.749			41.749
	3M-6J-V2-P1-S1-D9	57.952	57.952				57.952
	3M-6J-V2-P1-S1-D10	64.9	64.717				64.717
GROUP 6	3M-6J-V2-P1-S2-D1	48.761	48.225	48.225			48.225
	3M-6J-V2-P1-S2-D2	41.825	40.496	39.979	39.979		39.979
	3M-6J-V2-P1-S2-D3	66.881	66.881				66.881
	3M-6J-V2-P1-S2-D4	55.182	54.73				54.730
	3M-6J-V2-P1-S2-D5	62.388	62.388				62.388
	3M-6J-V2-P1-S2-D6	43.873	43.873				43.873
	3M-6J-V2-P1-S2-D7	46.308	44.885	44.885			44.885
	3M-6J-V2-P1-S2-D8	42.618	42.618				42.618
	3M-6J-V2-P1-S2-D9	57.952	57.952				57.952
	3M-6J-V2-P1-S2-D10	62.894	62.894				62.894
GROUP 7	3M-6J-V2-P2-S1-D1	51.934	50.079	50.079			50.079
	3M-6J-V2-P2-S1-D2	49.252	48.624	48.624			48.624
	3M-6J-V2-P2-S1-D3	58.579	58.579				58.579
	3M-6J-V2-P2-S1-D4	54.91	54.91				54.910
	3M-6J-V2-P2-S1-D5	56.336	56.336				56.336
	3M-6J-V2-P2-S1-D6	46.505	46.505				46.505
	3M-6J-V2-P2-S1-D7	51.077	51.077				51.077
	3M-6J-V2-P2-S1-D8	46.967	46.967				46.967
	3M-6J-V2-P2-S1-D9	55.169	54.954				54.954
	3M-6J-V2-P2-S1-D10	55.759	55.759				55.759
GROUP 8	3M-6J-V2-P2-S2-D1	49.411	49.411				49.411
	3M-6J-V2-P2-S2-D2	48.624	48.624				48.624
	3M-6J-V2-P2-S2-D3	58.158	58.158				58.158
	3M-6J-V2-P2-S2-D4	55.182	55.182				55.182
	3M-6J-V2-P2-S2-D5	56.233	55.152	55.152			55.152
	3M-6J-V2-P2-S2-D6	47.761	47.383				47.383
	3M-6J-V2-P2-S2-D7	51.077	51.077				51.077
	3M-6J-V2-P2-S2-D8	47.97	47.97				47.970
	3M-6J-V2-P2-S2-D9	56.343	55.857				55.857
	3M-6J-V2-P2-S2-D10	55.932	55.932				55.932

*H is the reputation time

		H 100 C_{max}	H 200 C_{max}	H 400 C_{max}	H 800 C_{max}	H 1600 C_{max}	H-Best C_{max}
GROUP 9	3M-12J-V1-P1-S1-D1	117.347	116.857				116.857
	3M-12J-V1-P1-S1-D2	78.175	76.418	76.418			76.418
	3M-12J-V1-P1-S1-D3	116.587	115.152	115.152			115.152
	3M-12J-V1-P1-S1-D4	84.496	84.496				84.496
	3M-12J-V1-P1-S1-D5	110.132	110.132				110.132
	3M-12J-V1-P1-S1-D6	93.708	91.404	91.07			91.070
	3M-12J-V1-P1-S1-D7	146.978	145.666				145.666
	3M-12J-V1-P1-S1-D8	87.579	87.579				87.579
	3M-12J-V1-P1-S1-D9	96.074	96.074				96.074
	3M-12J-V1-P1-S1-D10	126.388	125.334				125.334
GROUP 10	3M-12J-V1-P1-S2-D1	117.347	117.347				117.347
	3M-12J-V1-P1-S2-D2	79.053	78.681				78.681
	3M-12J-V1-P1-S2-D3	117.055	114.99	114.99			114.990
	3M-12J-V1-P1-S2-D4	86.124	85.365				85.365
	3M-12J-V1-P1-S2-D5	112.962	111.3	111.3			111.300
	3M-12J-V1-P1-S2-D6	91.404	89.868	89.868			89.868
	3M-12J-V1-P1-S2-D7	146.737	145.117	145.117			145.117
	3M-12J-V1-P1-S2-D8	88.13	88.13				88.130
	3M-12J-V1-P1-S2-D9	98.163	98.163				98.163
	3M-12J-V1-P1-S2-D10	125.638	124.43				124.430
GROUP 11	3M-12J-V1-P2-S1-D1	101.197	100.48				100.480
	3M-12J-V1-P2-S1-D2	81.27	81.27				81.270
	3M-12J-V1-P2-S1-D3	105.238	105.238				105.238
	3M-12J-V1-P2-S1-D4	92.807	92.807				92.807
	3M-12J-V1-P2-S1-D5	107.824	105.229	105.229			105.229
	3M-12J-V1-P2-S1-D6	95.135	95.135				95.135
	3M-12J-V1-P2-S1-D7	149.714	145.516	145.516			145.516
	3M-12J-V1-P2-S1-D8	96.962	96.524				96.524
	3M-12J-V1-P2-S1-D9	94.682	94.682				94.682
	3M-12J-V1-P2-S1-D10	162.612	161.433				161.433
GROUP 12	3M-12J-V1-P2-S2-D1	103.35	101.197	101.145			101.145
	3M-12J-V1-P2-S2-D2	81.688	80.81	80.81			80.810
	3M-12J-V1-P2-S2-D3	105.762	105.053				105.053
	3M-12J-V1-P2-S2-D4	93.045	91.688	91.507			91.507
	3M-12J-V1-P2-S2-D5	106.726	106.726				106.726
	3M-12J-V1-P2-S2-D6	97.062	93.509	93.509			93.509
	3M-12J-V1-P2-S2-D7	149.714	145.516	145.516			145.516
	3M-12J-V1-P2-S2-D8	100.09	99.048	96.333	96.333		96.333
	3M-12J-V1-P2-S2-D9	96.074	96.074				96.074
	3M-12J-V1-P2-S2-D10	166.541	162.798	162.798			162.798

*H is the reputation time

		H 100 C_{max}	H 200 C_{max}	H 400 C_{max}	H 800 C_{max}	H 1600 C_{max}	H-Best C_{max}
GROUP 13	3M-12J-V2-P1-S1-D1	111.744	111.744				111.744
	3M-12J-V2-P1-S1-D2	93.05	92.181				92.181
	3M-12J-V2-P1-S1-D3	128.268	124.146	124.146			124.146
	3M-12J-V2-P1-S1-D4	90.554	90.554				90.554
	3M-12J-V2-P1-S1-D5	108.957	107.156	105.784	105.784		105.784
	3M-12J-V2-P1-S1-D6	96.586	96.586				96.586
	3M-12J-V2-P1-S1-D7	110.194	110.194				110.194
	3M-12J-V2-P1-S1-D8	89.255	89.255				89.255
	3M-12J-V2-P1-S1-D9	106.993	106.993				106.993
	3M-12J-V2-P1-S1-D10	114.595	111.775	111.31			111.310
GROUP 14	3M-12J-V2-P1-S2-D1	110.836	110.836				110.836
	3M-12J-V2-P1-S2-D2	92.946	91.998	91.998			91.998
	3M-12J-V2-P1-S2-D3	127.174	127.174				127.174
	3M-12J-V2-P1-S2-D4	91.732	91.732				91.732
	3M-12J-V2-P1-S2-D5	109.933	106.821	106.821			106.821
	3M-12J-V2-P1-S2-D6	97.196	97.196				97.196
	3M-12J-V2-P1-S2-D7	110.224	110.224				110.224
	3M-12J-V2-P1-S2-D8	91.117	88.241	88.241			88.241
	3M-12J-V2-P1-S2-D9	108.488	107.877				107.877
	3M-12J-V2-P1-S2-D10	114.291	112.304	110.316	110.316		110.316
GROUP 15	3M-12J-V2-P2-S1-D1	107.391	106.521				106.521
	3M-12J-V2-P2-S1-D2	96.972	96.529				96.529
	3M-12J-V2-P2-S1-D3	114.016	114.016				114.016
	3M-12J-V2-P2-S1-D4	97.1	97.1				97.100
	3M-12J-V2-P2-S1-D5	109.2	109.2				109.200
	3M-12J-V2-P2-S1-D6	102.905	101.969				101.969
	3M-12J-V2-P2-S1-D7	117.894	115.325	115.325			115.325
	3M-12J-V2-P2-S1-D8	98.795	98.795				98.795
	3M-12J-V2-P2-S1-D9	106.293	106.109				106.109
	3M-12J-V2-P2-S1-D10	126.027	126.027				126.027
GROUP 16	3M-12J-V2-P2-S2-D1	107.391	107.391				107.391
	3M-12J-V2-P2-S2-D2	95.791	95.791				95.791
	3M-12J-V2-P2-S2-D3	117.579	116.229	116.229			116.229
	3M-12J-V2-P2-S2-D4	99.988	99.115				99.115
	3M-12J-V2-P2-S2-D5	109.857	108.16	108.16			108.160
	3M-12J-V2-P2-S2-D6	100.879	100.879				100.879
	3M-12J-V2-P2-S2-D7	118.919	117.609	116.467			116.467
	3M-12J-V2-P2-S2-D8	97.369	97.369				97.369
	3M-12J-V2-P2-S2-D9	105.24	104.714				104.714
	3M-12J-V2-P2-S2-D10	127.379	126.697				126.697

*H is the reputation time

		H 100 C_{max}	H 200 C_{max}	H 400 C_{max}	H 800 C_{max}	H 1600 C_{max}	H-Best C_{max}
GROUP 17	5M-10J-V1-P1-S1-D1	58.964	58.964				58.964
	5M-10J-V1-P1-S1-D2	46.554	45.714	45.714			45.714
	5M-10J-V1-P1-S1-D3	68.52	68.52				68.520
	5M-10J-V1-P1-S1-D4	47.043	43.072	43.072			43.072
	5M-10J-V1-P1-S1-D5	58.142	57.905				57.905
	5M-10J-V1-P1-S1-D6	59.801	59.801				59.801
	5M-10J-V1-P1-S1-D7	55.545	55.545				55.545
	5M-10J-V1-P1-S1-D8	46.415	46.415				46.415
	5M-10J-V1-P1-S1-D9	63.583	62.859	62.859			62.859
	5M-10J-V1-P1-S1-D10	67.651	67.651				67.651
GROUP 18	5M-10J-V1-P1-S2-D1	62.459	61.109	58.936	58.936		58.936
	5M-10J-V1-P1-S2-D2	44.798	44.798				44.798
	5M-10J-V1-P1-S2-D3	65.303	65.303				65.303
	5M-10J-V1-P1-S2-D4	44.339	44.339				44.339
	5M-10J-V1-P1-S2-D5	60.175	60.175				60.175
	5M-10J-V1-P1-S2-D6	67.064	60.473	60.163			60.163
	5M-10J-V1-P1-S2-D7	57.459	57.459				57.459
	5M-10J-V1-P1-S2-D8	48.262	47.237	47.237			47.237
	5M-10J-V1-P1-S2-D9	65.448	61.792	61.792			61.792
	5M-10J-V1-P1-S2-D10	68	68				68.000
GROUP 19	5M-10J-V1-P2-S1-D1	62.137	62.137				62.137
	5M-10J-V1-P2-S1-D2	44.021	44.021				44.021
	5M-10J-V1-P2-S1-D3	60.601	60.601				60.601
	5M-10J-V1-P2-S1-D4	49.664	49.664				49.664
	5M-10J-V1-P2-S1-D5	63.979	63.979				63.979
	5M-10J-V1-P2-S1-D6	63.232	63.232				63.232
	5M-10J-V1-P2-S1-D7	82.472	82.472				82.472
	5M-10J-V1-P2-S1-D8	55.173	55.173				55.173
	5M-10J-V1-P2-S1-D9	60.788	59.358	57.86	57.86		57.860
	5M-10J-V1-P2-S1-D10	69.334	69.334				69.334
GROUP 20	5M-10J-V1-P2-S2-D1	63.253	63.253				63.253
	5M-10J-V1-P2-S2-D2	46.554	46.554				46.554
	5M-10J-V1-P2-S2-D3	62.41	62.41				62.410
	5M-10J-V1-P2-S2-D4	49.983	49.983				49.983
	5M-10J-V1-P2-S2-D5	65.011	65.011				65.011
	5M-10J-V1-P2-S2-D6	63.462	63.462				63.462
	5M-10J-V1-P2-S2-D7	80.645	80.645				80.645
	5M-10J-V1-P2-S2-D8	54.557	54.557				54.557
	5M-10J-V1-P2-S2-D9	58.21	58.21				58.210
	5M-10J-V1-P2-S2-D10	70.223	66.443	66.443			66.443

*H is the reputation time

		H 100 Cmax	H 200 Cmax	H 400 Cmax	H 800 Cmax	H 1600 Cmax	H-Best Cmax
GROUP 21	5M-10J-V2-P1-S1-D1	61.492	60.215	58.173	56.132	56.132	56.132
	5M-10J-V2-P1-S1-D2	49.958	49.491				49.491
	5M-10J-V2-P1-S1-D3	76.605	70.92	70.92			70.920
	5M-10J-V2-P1-S1-D4	45.335	45.335				45.335
	5M-10J-V2-P1-S1-D5	54.509	53.128	52.984			52.984
	5M-10J-V2-P1-S1-D6	56.036	56.036				56.036
	5M-10J-V2-P1-S1-D7	52.542	52.283				52.283
	5M-10J-V2-P1-S1-D8	46.011	46.011				46.011
	5M-10J-V2-P1-S1-D9	60.624	60.624				60.624
	5M-10J-V2-P1-S1-D10	67.581	65.963	65.963			65.963
GROUP 22	5M-10J-V2-P1-S2-D1	59.96	58.241	58.241			58.241
	5M-10J-V2-P1-S2-D2	49.491	47.83	47.83			47.830
	5M-10J-V2-P1-S2-D3	72.889	72.889				72.889
	5M-10J-V2-P1-S2-D4	43.765	43.765				43.765
	5M-10J-V2-P1-S2-D5	56.729	53.128	51.327	51.327		51.327
	5M-10J-V2-P1-S2-D6	58.373	57.567	57.567			57.567
	5M-10J-V2-P1-S2-D7	53.309	52.05	51.399	50.965		50.965
	5M-10J-V2-P1-S2-D8	45.642	45.081	45.081			45.081
	5M-10J-V2-P1-S2-D9	61.466	60.023	60.023			60.023
	5M-10J-V2-P1-S2-D10	71.556	67.581	67.581			67.581
GROUP 23	5M-10J-V2-P2-S1-D1	61.492	61.492				61.492
	5M-10J-V2-P2-S1-D2	52.282	51.12	50.267	48.37	48.37	48.370
	5M-10J-V2-P2-S1-D3	62.578	60.571	60.571			60.571
	5M-10J-V2-P2-S1-D4	50.691	48.284	48.284			48.284
	5M-10J-V2-P2-S1-D5	60.562	60.562				60.562
	5M-10J-V2-P2-S1-D6	61.165	56.038	56.038			56.038
	5M-10J-V2-P2-S1-D7	62.001	59.949	59.949			59.949
	5M-10J-V2-P2-S1-D8	49.608	49.608				49.608
	5M-10J-V2-P2-S1-D9	57.987	57.987				57.987
	5M-10J-V2-P2-S1-D10	61.183	58.636	58.636			58.636
GROUP 24	5M-10J-V2-P2-S2-D1	58.241	58.241				58.241
	5M-10J-V2-P2-S2-D2	51.308	51.308				51.308
	5M-10J-V2-P2-S2-D3	64.134	62.055	62.055			62.055
	5M-10J-V2-P2-S2-D4	47.299	47.299				47.299
	5M-10J-V2-P2-S2-D5	61.197	61.197				61.197
	5M-10J-V2-P2-S2-D6	58.373	55.768	55.768			55.768
	5M-10J-V2-P2-S2-D7	63.149	62.255	62.255			62.255
	5M-10J-V2-P2-S2-D8	50.713	50.713				50.713
	5M-10J-V2-P2-S2-D9	58.336	58.336				58.336
	5M-10J-V2-P2-S2-D10	57.902	57.902				57.902

*H is the reputation time

		H 100 C_{max}	H 200 C_{max}	H 400 C_{max}	H 800 C_{max}	H 1600 C_{max}	H-Best C_{max}
GROUP 25	5M-20J-V1-P1-S1-D1	127.859	127.859				127.859
	5M-20J-V1-P1-S1-D2	98.466	98.378				98.378
	5M-20J-V1-P1-S1-D3	128.175	127.571				127.571
	5M-20J-V1-P1-S1-D4	110.342	106.338	106.076			106.076
	5M-20J-V1-P1-S1-D5	107.912	104.56	103.507	103.507		103.507
	5M-20J-V1-P1-S1-D6	95.208	93.69	93.581			93.581
	5M-20J-V1-P1-S1-D7	105.083	104.2				104.200
	5M-20J-V1-P1-S1-D8	94.164	93.243				93.243
	5M-20J-V1-P1-S1-D9	135.432	130.184	130.184			130.184
	5M-20J-V1-P1-S1-D10	104.355	104.355				104.355
GROUP 26	5M-20J-V1-P1-S2-D1	129.616	129.616				129.616
	5M-20J-V1-P1-S2-D2	99.535	97.221	97.221			97.221
	5M-20J-V1-P1-S2-D3	129.108	126.888	126.888			126.888
	5M-20J-V1-P1-S2-D4	110.277	110.277				110.277
	5M-20J-V1-P1-S2-D5	101.432	101.432				101.432
	5M-20J-V1-P1-S2-D6	93.731	93.731				93.731
	5M-20J-V1-P1-S2-D7	103.776	103.776				103.776
	5M-20J-V1-P1-S2-D8	95.422	95.181				95.181
	5M-20J-V1-P1-S2-D9	137.491	136.854				136.854
	5M-20J-V1-P1-S2-D10	104.399	104.399				104.399
GROUP 27	5M-20J-V1-P2-S1-D1	107.246	107.246				107.246
	5M-20J-V1-P2-S1-D2	97.221	91.81	91.81			91.810
	5M-20J-V1-P2-S1-D3	119.879	117.195	115.433	115.433		115.433
	5M-20J-V1-P2-S1-D4	115.528	113.12	113.12			113.120
	5M-20J-V1-P2-S1-D5	127.2	124.83	124.442			124.442
	5M-20J-V1-P2-S1-D6	99.095	94.036	94.036			94.036
	5M-20J-V1-P2-S1-D7	101.803	101.803				101.803
	5M-20J-V1-P2-S1-D8	143.016	143.016				143.016
	5M-20J-V1-P2-S1-D9	132.388	132.388				132.388
	5M-20J-V1-P2-S1-D10	120.559	119.514				119.514
GROUP 28	5M-20J-V1-P2-S2-D1	107.747	107.242				107.242
	5M-20J-V1-P2-S2-D2	94.204	94.204				94.204
	5M-20J-V1-P2-S2-D3	117.64	117.64				117.640
	5M-20J-V1-P2-S2-D4	114.435	114.215				114.215
	5M-20J-V1-P2-S2-D5	121.385	121.385				121.385
	5M-20J-V1-P2-S2-D6	99.679	95.871	94.977			94.977
	5M-20J-V1-P2-S2-D7	98.371	98.371				98.371
	5M-20J-V1-P2-S2-D8	155.772	143.553	143.553			143.553
	5M-20J-V1-P2-S2-D9	131.434	131.434				131.434
	5M-20J-V1-P2-S2-D10	121.626	121.626				121.626

*H is the reputation time

		H 100 C_{max}	H 200 C_{max}	H 400 C_{max}	H 800 C_{max}	H 1600 C_{max}	H-Best C_{max}
GROUP 29	5M-20J-V2-P1-S1-D1	114.859	113.226	113.226			113.226
	5M-20J-V2-P1-S1-D2	109.78	109.78				109.780
	5M-20J-V2-P1-S1-D3	131.936	131.936				131.936
	5M-20J-V2-P1-S1-D4	102.601	101.51	101.51			101.510
	5M-20J-V2-P1-S1-D5	104.89	104.89				104.890
	5M-20J-V2-P1-S1-D6	104.804	104.804				104.804
	5M-20J-V2-P1-S1-D7	101.421	101.421				101.421
	5M-20J-V2-P1-S1-D8	101.858	101.858				101.858
	5M-20J-V2-P1-S1-D9	123.724	123.724				123.724
	5M-20J-V2-P1-S1-D10	108.637	108.637				108.637
GROUP 30	5M-20J-V2-P1-S2-D1	117.779	117.719				117.719
	5M-20J-V2-P1-S2-D2	109.932	109.932				109.932
	5M-20J-V2-P1-S2-D3	135.985	133.95	133.433			133.433
	5M-20J-V2-P1-S2-D4	101.435	101.435				101.435
	5M-20J-V2-P1-S2-D5	106.122	106.122				106.122
	5M-20J-V2-P1-S2-D6	105.984	105.984				105.984
	5M-20J-V2-P1-S2-D7	99.24	99.24				99.240
	5M-20J-V2-P1-S2-D8	102.796	102.796				102.796
	5M-20J-V2-P1-S2-D9	124.207	124.207				124.207
	5M-20J-V2-P1-S2-D10	109.607	109.607				109.607
GROUP 31	5M-20J-V2-P2-S1-D1	110.27	107.833	107.833			107.833
	5M-20J-V2-P2-S1-D2	108.423	107.381				107.381
	5M-20J-V2-P2-S1-D3	122.874	122.874				122.874
	5M-20J-V2-P2-S1-D4	107.75	107.689				107.689
	5M-20J-V2-P2-S1-D5	119.121	119.121				119.121
	5M-20J-V2-P2-S1-D6	105.828	102.968	102.968			102.968
	5M-20J-V2-P2-S1-D7	103.719	102.431	101.016	101.016		101.016
	5M-20J-V2-P2-S1-D8	120.088	120.088				120.088
	5M-20J-V2-P2-S1-D9	121.139	118.777	118.777			118.777
	5M-20J-V2-P2-S1-D10	117.481	117.481				117.481
GROUP 32	5M-20J-V2-P2-S2-D1	114.301	113.212				113.212
	5M-20J-V2-P2-S2-D2	108.007	108.007				108.007
	5M-20J-V2-P2-S2-D3	122.477	122.477				122.477
	5M-20J-V2-P2-S2-D4	107.689	107.689				107.689
	5M-20J-V2-P2-S2-D5	117.793	117.793				117.793
	5M-20J-V2-P2-S2-D6	107.453	105.828	103.868	103.868		103.868
	5M-20J-V2-P2-S2-D7	103.719	103.719				103.719
	5M-20J-V2-P2-S2-D8	121.496	121.496				121.496
	5M-20J-V2-P2-S2-D9	119.383	117.212	117.212			117.212
	5M-20J-V2-P2-S2-D10	117.765	115.522	115.522			115.522

*H is the reputation time

APPENDIX 3 – COMPARISON FOR INITIAL HEURISTIC AND IMPROVED HEURISTIC SOLUTIONS

		Initial Heuristic C_{max}	Improved Heuristic C_{max}	% Gap
GROUP 1	3M-6J-V1-P1-S1-D1	104.591	46.611	124.39%
	3M-6J-V1-P1-S1-D2	108.263	52.265	107.14%
	3M-6J-V1-P1-S1-D3	64.764	64.764	0.00%
	3M-6J-V1-P1-S1-D4	67.331	67.331	0.00%
	3M-6J-V1-P1-S1-D5	72.091	72.091	0.00%
	3M-6J-V1-P1-S1-D6	40.941	40.941	0.00%
	3M-6J-V1-P1-S1-D7	44.33	44.330	0.00%
	3M-6J-V1-P1-S1-D8	92.278	44.693	106.47%
	3M-6J-V1-P1-S1-D9	122.424	57.809	111.77%
	3M-6J-V1-P1-S1-D10	188.359	56.346	234.29%
GROUP 2	3M-6J-V1-P1-S2-D1	64.798	48.342	34.04%
	3M-6J-V1-P1-S2-D2	94.072	51.890	81.29%
	3M-6J-V1-P1-S2-D3	62.41	62.410	0.00%
	3M-6J-V1-P1-S2-D4	67.331	67.331	0.00%
	3M-6J-V1-P1-S2-D5	217.449	69.751	211.75%
	3M-6J-V1-P1-S2-D6	71.647	44.353	61.54%
	3M-6J-V1-P1-S2-D7	95.939	45.254	112.00%
	3M-6J-V1-P1-S2-D8	85.443	47.848	78.57%
	3M-6J-V1-P1-S2-D9	223.128	57.839	285.77%
	3M-6J-V1-P1-S2-D10	127.461	56.664	124.94%
GROUP 3	3M-6J-V1-P2-S1-D1	73.027	49.370	47.92%
	3M-6J-V1-P2-S1-D2	158.661	65.991	140.43%
	3M-6J-V1-P2-S1-D3	123.976	55.438	123.63%
	3M-6J-V1-P2-S1-D4	93.387	72.422	28.95%
	3M-6J-V1-P2-S1-D5	75.238	52.066	44.51%
	3M-6J-V1-P2-S1-D6	46.47	46.470	0.00%
	3M-6J-V1-P2-S1-D7	66.495	50.795	30.91%
	3M-6J-V1-P2-S1-D8	48.695	48.695	0.00%
	3M-6J-V1-P2-S1-D9	83.421	56.346	48.05%
	3M-6J-V1-P2-S1-D10	99.729	49.493	101.50%
GROUP 4	3M-6J-V1-P2-S2-D1	51.427	51.427	0.00%
	3M-6J-V1-P2-S2-D2	94.072	65.331	43.99%
	3M-6J-V1-P2-S2-D3	92.397	55.174	67.46%
	3M-6J-V1-P2-S2-D4	72.422	72.422	0.00%
	3M-6J-V1-P2-S2-D5	78.766	54.077	45.66%
	3M-6J-V1-P2-S2-D6	92.96	46.400	100.34%
	3M-6J-V1-P2-S2-D7	77.424	49.952	55.00%
	3M-6J-V1-P2-S2-D8	50.84	50.840	0.00%
	3M-6J-V1-P2-S2-D9	165.865	56.346	194.37%
	3M-6J-V1-P2-S2-D10	62.708	49.109	27.69%

		Initial Heuristic C_{max}	Improvement Heuristic C_{max}	% Gap
GROUP 5	3M-6J-V2-P1-S1-D1	77.647	46.686	66.32%
	3M-6J-V2-P1-S1-D2	40.664	40.664	0.00%
	3M-6J-V2-P1-S1-D3	108.371	67.851	59.72%
	3M-6J-V2-P1-S1-D4	94.638	53.590	76.60%
	3M-6J-V2-P1-S1-D5	96.246	62.388	54.27%
	3M-6J-V2-P1-S1-D6	80.901	42.995	88.16%
	3M-6J-V2-P1-S1-D7	84.26	44.407	89.74%
	3M-6J-V2-P1-S1-D8	78.984	41.749	89.19%
	3M-6J-V2-P1-S1-D9	104.507	57.952	80.33%
	3M-6J-V2-P1-S1-D10	113.939	64.717	76.06%
GROUP 6	3M-6J-V2-P1-S2-D1	82.997	48.225	72.10%
	3M-6J-V2-P1-S2-D2	72.395	39.979	81.08%
	3M-6J-V2-P1-S2-D3	97.761	66.881	46.17%
	3M-6J-V2-P1-S2-D4	79.815	54.730	45.83%
	3M-6J-V2-P1-S2-D5	103.979	62.388	66.67%
	3M-6J-V2-P1-S2-D6	56.157	43.873	28.00%
	3M-6J-V2-P1-S2-D7	92.698	44.885	106.52%
	3M-6J-V2-P1-S2-D8	80.066	42.618	87.87%
	3M-6J-V2-P1-S2-D9	104.507	57.952	80.33%
	3M-6J-V2-P1-S2-D10	112.116	62.894	78.26%
GROUP 7	3M-6J-V2-P2-S1-D1	70.482	50.079	40.74%
	3M-6J-V2-P2-S1-D2	48.624	48.624	0.00%
	3M-6J-V2-P2-S1-D3	119.111	58.579	103.33%
	3M-6J-V2-P2-S1-D4	91.127	54.910	65.96%
	3M-6J-V2-P2-S1-D5	85.432	56.336	51.65%
	3M-6J-V2-P2-S1-D6	46.505	46.505	0.00%
	3M-6J-V2-P2-S1-D7	85.737	51.077	67.86%
	3M-6J-V2-P2-S1-D8	82.234	46.967	75.09%
	3M-6J-V2-P2-S1-D9	78.934	54.954	43.64%
	3M-6J-V2-P2-S1-D10	55.759	55.759	0.00%
GROUP 8	3M-6J-V2-P2-S2-D1	68.627	49.411	38.89%
	3M-6J-V2-P2-S2-D2	70.048	48.624	44.06%
	3M-6J-V2-P2-S2-D3	120.941	58.158	107.95%
	3M-6J-V2-P2-S2-D4	100.51	55.182	82.14%
	3M-6J-V2-P2-S2-D5	89.584	55.152	62.43%
	3M-6J-V2-P2-S2-D6	47.383	47.383	0.00%
	3M-6J-V2-P2-S2-D7	67.495	51.077	32.14%
	3M-6J-V2-P2-S2-D8	75.381	47.970	57.14%
	3M-6J-V2-P2-S2-D9	55.857	55.857	0.00%
	3M-6J-V2-P2-S2-D10	108.573	55.932	94.12%

		Initial Heuristic C_{max}	Improvement Heuristic C_{max}	% Gap
GROUP 9	3M-12J-V1-P1-S1-D1	317.973	116.857	172.10%
	3M-12J-V1-P1-S1-D2	114.188	76.418	49.43%
	3M-12J-V1-P1-S1-D3	139.285	115.152	20.96%
	3M-12J-V1-P1-S1-D4	117.771	84.496	39.38%
	3M-12J-V1-P1-S1-D5	250.841	110.132	127.76%
	3M-12J-V1-P1-S1-D6	143.652	91.070	57.74%
	3M-12J-V1-P1-S1-D7	145.666	145.666	0.00%
	3M-12J-V1-P1-S1-D8	109.474	87.579	25.00%
	3M-12J-V1-P1-S1-D9	199.035	96.074	107.17%
	3M-12J-V1-P1-S1-D10	175.169	125.334	39.76%
GROUP 10	3M-12J-V1-P1-S2-D1	193.055	117.347	64.52%
	3M-12J-V1-P1-S2-D2	86.08	78.681	9.40%
	3M-12J-V1-P1-S2-D3	161.784	114.990	40.69%
	3M-12J-V1-P1-S2-D4	121.723	85.365	42.59%
	3M-12J-V1-P1-S2-D5	147.613	111.300	32.63%
	3M-12J-V1-P1-S2-D6	135.954	89.868	51.28%
	3M-12J-V1-P1-S2-D7	303.258	145.117	108.97%
	3M-12J-V1-P1-S2-D8	133.991	88.130	52.04%
	3M-12J-V1-P1-S2-D9	151.075	98.163	53.90%
	3M-12J-V1-P1-S2-D10	136.001	124.430	9.30%
GROUP 11	3M-12J-V1-P2-S1-D1	114.834	100.480	14.29%
	3M-12J-V1-P2-S1-D2	106.283	81.270	30.78%
	3M-12J-V1-P2-S1-D3	170.238	105.238	61.76%
	3M-12J-V1-P2-S1-D4	157.91	92.807	70.15%
	3M-12J-V1-P2-S1-D5	157.113	105.229	49.31%
	3M-12J-V1-P2-S1-D6	203.184	95.135	113.57%
	3M-12J-V1-P2-S1-D7	261.116	145.516	79.44%
	3M-12J-V1-P2-S1-D8	147.008	96.524	52.30%
	3M-12J-V1-P2-S1-D9	107.91	94.682	13.97%
	3M-12J-V1-P2-S1-D10	333.282	161.433	106.45%
GROUP 12	3M-12J-V1-P2-S2-D1	101.145	101.145	0.00%
	3M-12J-V1-P2-S2-D2	99.256	80.810	22.83%
	3M-12J-V1-P2-S2-D3	132.026	105.053	25.68%
	3M-12J-V1-P2-S2-D4	189.77	91.507	107.38%
	3M-12J-V1-P2-S2-D5	205.324	106.726	92.38%
	3M-12J-V1-P2-S2-D6	131.726	93.509	40.87%
	3M-12J-V1-P2-S2-D7	173.5	145.516	19.23%
	3M-12J-V1-P2-S2-D8	125.113	96.333	29.88%
	3M-12J-V1-P2-S2-D9	131.891	96.074	37.28%
	3M-12J-V1-P2-S2-D10	267.32	162.798	64.20%

		Initial Heuristic C_{max}	Improvement Heuristic C_{max}	% Gap
GROUP 13	3M-12J-V2-P1-S1-D1	158.077	111.744	41.46%
	3M-12J-V2-P1-S1-D2	178.581	92.181	93.73%
	3M-12J-V2-P1-S1-D3	241.227	124.146	94.31%
	3M-12J-V2-P1-S1-D4	137.468	90.554	51.81%
	3M-12J-V2-P1-S1-D5	161.788	105.784	52.94%
	3M-12J-V2-P1-S1-D6	96.586	96.586	0.00%
	3M-12J-V2-P1-S1-D7	189.26	110.194	71.75%
	3M-12J-V2-P1-S1-D8	128.53	89.255	44.00%
	3M-12J-V2-P1-S1-D9	130.867	106.993	22.31%
	3M-12J-V2-P1-S1-D10	160	111.310	43.74%
GROUP 14	3M-12J-V2-P1-S2-D1	133.954	110.836	20.86%
	3M-12J-V2-P1-S2-D2	91.998	91.998	0.00%
	3M-12J-V2-P1-S2-D3	188.742	127.174	48.41%
	3M-12J-V2-P1-S2-D4	177.835	91.732	93.86%
	3M-12J-V2-P1-S2-D5	141.95	106.821	32.89%
	3M-12J-V2-P1-S2-D6	142.733	97.196	46.85%
	3M-12J-V2-P1-S2-D7	168.78	110.224	53.12%
	3M-12J-V2-P1-S2-D8	155.911	88.241	76.69%
	3M-12J-V2-P1-S2-D9	146.284	107.877	35.60%
	3M-12J-V2-P1-S2-D10	231.564	110.316	109.91%
GROUP 15	3M-12J-V2-P2-S1-D1	194.709	106.521	82.79%
	3M-12J-V2-P2-S1-D2	139.141	96.529	44.14%
	3M-12J-V2-P2-S1-D3	163.899	114.016	43.75%
	3M-12J-V2-P2-S1-D4	170.442	97.100	75.53%
	3M-12J-V2-P2-S1-D5	179.92	109.200	64.76%
	3M-12J-V2-P2-S1-D6	191.777	101.969	88.07%
	3M-12J-V2-P2-S1-D7	206.672	115.325	79.21%
	3M-12J-V2-P2-S1-D8	117.44	98.795	18.87%
	3M-12J-V2-P2-S1-D9	161.816	106.109	52.50%
	3M-12J-V2-P2-S1-D10	126.027	126.027	0.00%
GROUP 16	3M-12J-V2-P2-S2-D1	203.735	107.391	89.71%
	3M-12J-V2-P2-S2-D2	115.214	95.791	20.28%
	3M-12J-V2-P2-S2-D3	151.689	116.229	30.51%
	3M-12J-V2-P2-S2-D4	127.831	99.115	28.97%
	3M-12J-V2-P2-S2-D5	217.36	108.160	100.96%
	3M-12J-V2-P2-S2-D6	148.546	100.879	47.25%
	3M-12J-V2-P2-S2-D7	182.693	116.467	56.86%
	3M-12J-V2-P2-S2-D8	211.978	97.369	117.71%
	3M-12J-V2-P2-S2-D9	165.09	104.714	57.66%
	3M-12J-V2-P2-S2-D10	299.028	126.697	136.02%

		Initial Heuristic C_{max}	Improvement Heuristic C_{max}	% Gap
GROUP 17	5M-10J-V1-P1-S1-D1	174.128	58.964	195.31%
	5M-10J-V1-P1-S1-D2	108.263	45.714	136.83%
	5M-10J-V1-P1-S1-D3	124.323	68.520	81.44%
	5M-10J-V1-P1-S1-D4	180.729	43.072	319.60%
	5M-10J-V1-P1-S1-D5	98.779	57.905	70.59%
	5M-10J-V1-P1-S1-D6	178.199	59.801	197.99%
	5M-10J-V1-P1-S1-D7	212.677	55.545	282.89%
	5M-10J-V1-P1-S1-D8	67.769	46.415	46.01%
	5M-10J-V1-P1-S1-D9	100.543	62.859	59.95%
	5M-10J-V1-P1-S1-D10	168.123	67.651	148.52%
GROUP 18	5M-10J-V1-P1-S2-D1	58.936	58.936	0.00%
	5M-10J-V1-P1-S2-D2	106.396	44.798	137.50%
	5M-10J-V1-P1-S2-D3	126.347	65.303	93.48%
	5M-10J-V1-P1-S2-D4	138.084	44.339	211.43%
	5M-10J-V1-P1-S2-D5	98.647	60.175	63.93%
	5M-10J-V1-P1-S2-D6	112.253	60.163	86.58%
	5M-10J-V1-P1-S2-D7	57.459	57.459	0.00%
	5M-10J-V1-P1-S2-D8	85.824	47.237	81.69%
	5M-10J-V1-P1-S2-D9	117.315	61.792	89.85%
	5M-10J-V1-P1-S2-D10	109.934	68.000	61.67%
GROUP 19	5M-10J-V1-P2-S1-D1	99.568	62.137	60.24%
	5M-10J-V1-P2-S1-D2	169.86	44.021	285.86%
	5M-10J-V1-P2-S1-D3	127.533	60.601	110.45%
	5M-10J-V1-P2-S1-D4	116.311	49.664	134.20%
	5M-10J-V1-P2-S1-D5	63.979	63.979	0.00%
	5M-10J-V1-P2-S1-D6	329.571	63.232	421.21%
	5M-10J-V1-P2-S1-D7	100.681	82.472	22.08%
	5M-10J-V1-P2-S1-D8	76.191	55.173	38.09%
	5M-10J-V1-P2-S1-D9	123.964	57.860	114.25%
	5M-10J-V1-P2-S1-D10	154.632	69.334	123.02%
GROUP 20	5M-10J-V1-P2-S2-D1	174.128	63.253	175.29%
	5M-10J-V1-P2-S2-D2	76.53	46.554	64.39%
	5M-10J-V1-P2-S2-D3	120.714	62.410	93.42%
	5M-10J-V1-P2-S2-D4	108.947	49.983	117.97%
	5M-10J-V1-P2-S2-D5	155.377	65.011	139.00%
	5M-10J-V1-P2-S2-D6	148.974	63.462	134.75%
	5M-10J-V1-P2-S2-D7	105.218	80.645	30.47%
	5M-10J-V1-P2-S2-D8	73.442	54.557	34.62%
	5M-10J-V1-P2-S2-D9	149.752	58.210	157.26%
	5M-10J-V1-P2-S2-D10	110.667	66.443	66.56%

		Initial Heuristic C_{max}	Improvement Heuristic C_{max}	% Gap
GROUP 21	5M-10J-V2-P1-S1-D1	76.019	56.132	35.43%
	5M-10J-V2-P1-S1-D2	49.491	49.491	0.00%
	5M-10J-V2-P1-S1-D3	150.24	70.920	111.84%
	5M-10J-V2-P1-S1-D4	78.465	45.335	73.08%
	5M-10J-V2-P1-S1-D5	112.457	52.984	112.25%
	5M-10J-V2-P1-S1-D6	123.703	56.036	120.76%
	5M-10J-V2-P1-S1-D7	94.315	52.283	80.39%
	5M-10J-V2-P1-S1-D8	63.898	46.011	38.88%
	5M-10J-V2-P1-S1-D9	96.258	60.624	58.78%
	5M-10J-V2-P1-S1-D10	105.783	65.963	60.37%
GROUP 22	5M-10J-V2-P1-S2-D1	105.904	58.241	81.84%
	5M-10J-V2-P1-S2-D2	94.107	47.830	96.75%
	5M-10J-V2-P1-S2-D3	145.779	72.889	100.00%
	5M-10J-V2-P1-S2-D4	92.203	43.765	110.68%
	5M-10J-V2-P1-S2-D5	51.327	51.327	0.00%
	5M-10J-V2-P1-S2-D6	93.546	57.567	62.50%
	5M-10J-V2-P1-S2-D7	79.928	50.965	56.83%
	5M-10J-V2-P1-S2-D8	105.509	45.081	134.04%
	5M-10J-V2-P1-S2-D9	127.57	60.023	112.54%
	5M-10J-V2-P1-S2-D10	152.9	67.581	126.25%
GROUP 23	5M-10J-V2-P2-S1-D1	133.233	61.492	116.67%
	5M-10J-V2-P2-S1-D2	119.139	48.370	146.31%
	5M-10J-V2-P2-S1-D3	146.974	60.571	142.65%
	5M-10J-V2-P2-S1-D4	73.654	48.284	52.54%
	5M-10J-V2-P2-S1-D5	184.714	60.562	205.00%
	5M-10J-V2-P2-S1-D6	135.424	56.038	141.66%
	5M-10J-V2-P2-S1-D7	90.808	59.949	51.48%
	5M-10J-V2-P2-S1-D8	71.881	49.608	44.90%
	5M-10J-V2-P2-S1-D9	162.362	57.987	180.00%
	5M-10J-V2-P2-S1-D10	126.217	58.636	115.26%
GROUP 24	5M-10J-V2-P2-S2-D1	111.822	58.241	92.00%
	5M-10J-V2-P2-S2-D2	75.153	51.308	46.47%
	5M-10J-V2-P2-S2-D3	115.062	62.055	85.42%
	5M-10J-V2-P2-S2-D4	113.309	47.299	139.56%
	5M-10J-V2-P2-S2-D5	222.86	61.197	264.17%
	5M-10J-V2-P2-S2-D6	55.768	55.768	0.00%
	5M-10J-V2-P2-S2-D7	83.007	62.255	33.33%
	5M-10J-V2-P2-S2-D8	80.127	50.713	58.00%
	5M-10J-V2-P2-S2-D9	125.823	58.336	115.69%
	5M-10J-V2-P2-S2-D10	103.246	57.902	78.31%

		Initial Heuristic C_{max}	Improvement Heuristic C_{max}	% Gap
GROUP 25	5M-20J-V1-P1-S1-D1	187.637	127.859	46.75%
	5M-20J-V1-P1-S1-D2	124.723	98.378	26.78%
	5M-20J-V1-P1-S1-D3	267.439	127.571	109.64%
	5M-20J-V1-P1-S1-D4	131.5	106.076	23.97%
	5M-20J-V1-P1-S1-D5	211.12	103.507	103.97%
	5M-20J-V1-P1-S1-D6	105.875	93.581	13.14%
	5M-20J-V1-P1-S1-D7	147.483	104.200	41.54%
	5M-20J-V1-P1-S1-D8	129.874	93.243	39.29%
	5M-20J-V1-P1-S1-D9	228.256	130.184	75.33%
	5M-20J-V1-P1-S1-D10	192.696	104.355	84.65%
GROUP 26	5M-20J-V1-P1-S2-D1	273.777	129.616	111.22%
	5M-20J-V1-P1-S2-D2	181.71	97.221	86.90%
	5M-20J-V1-P1-S2-D3	171.045	126.888	34.80%
	5M-20J-V1-P1-S2-D4	164.102	110.277	48.81%
	5M-20J-V1-P1-S2-D5	229.037	101.432	125.80%
	5M-20J-V1-P1-S2-D6	114.369	93.731	22.02%
	5M-20J-V1-P1-S2-D7	134.906	103.776	30.00%
	5M-20J-V1-P1-S2-D8	138.835	95.181	45.86%
	5M-20J-V1-P1-S2-D9	250.673	136.854	83.17%
	5M-20J-V1-P1-S2-D10	146.755	104.399	40.57%
GROUP 27	5M-20J-V1-P2-S1-D1	280.819	107.246	161.85%
	5M-20J-V1-P2-S1-D2	113.839	91.810	23.99%
	5M-20J-V1-P2-S1-D3	155.085	115.433	34.35%
	5M-20J-V1-P2-S1-D4	207.366	113.120	83.32%
	5M-20J-V1-P2-S1-D5	239.713	124.442	92.63%
	5M-20J-V1-P2-S1-D6	140.889	94.036	49.82%
	5M-20J-V1-P2-S1-D7	248.609	101.803	144.21%
	5M-20J-V1-P2-S1-D8	231.55	143.016	61.90%
	5M-20J-V1-P2-S1-D9	234.413	132.388	77.07%
	5M-20J-V1-P2-S1-D10	191.222	119.514	60.00%
GROUP 28	5M-20J-V1-P2-S2-D1	185.246	107.242	72.74%
	5M-20J-V1-P2-S2-D2	210.645	94.204	123.61%
	5M-20J-V1-P2-S2-D3	150.56	117.640	27.98%
	5M-20J-V1-P2-S2-D4	334.83	114.215	193.16%
	5M-20J-V1-P2-S2-D5	177.348	121.385	46.10%
	5M-20J-V1-P2-S2-D6	171.735	94.977	80.82%
	5M-20J-V1-P2-S2-D7	261.129	98.371	165.45%
	5M-20J-V1-P2-S2-D8	195.059	143.553	35.88%
	5M-20J-V1-P2-S2-D9	219.509	131.434	67.01%
	5M-20J-V1-P2-S2-D10	206.673	121.626	69.93%

		Initial Heuristic C_{max}	Improvement Heuristic C_{max}	% Gap
GROUP 29	5M-20J-V2-P1-S1-D1	224.284	113.226	98.09%
	5M-20J-V2-P1-S1-D2	189.455	109.780	72.58%
	5M-20J-V2-P1-S1-D3	301.136	131.936	128.24%
	5M-20J-V2-P1-S1-D4	192.872	101.510	90.00%
	5M-20J-V2-P1-S1-D5	198.587	104.890	89.33%
	5M-20J-V2-P1-S1-D6	132.876	104.804	26.79%
	5M-20J-V2-P1-S1-D7	140.624	101.421	38.65%
	5M-20J-V2-P1-S1-D8	176.79	101.858	73.57%
	5M-20J-V2-P1-S1-D9	228.692	123.724	84.84%
	5M-20J-V2-P1-S1-D10	192.23	108.637	76.95%
GROUP 30	5M-20J-V2-P1-S2-D1	171.387	117.719	45.59%
	5M-20J-V2-P1-S2-D2	177.89	109.932	61.82%
	5M-20J-V2-P1-S2-D3	180.279	133.433	35.11%
	5M-20J-V2-P1-S2-D4	177.22	101.435	74.71%
	5M-20J-V2-P1-S2-D5	203.045	106.122	91.33%
	5M-20J-V2-P1-S2-D6	150.02	105.984	41.55%
	5M-20J-V2-P1-S2-D7	127.498	99.240	28.47%
	5M-20J-V2-P1-S2-D8	145.651	102.796	41.69%
	5M-20J-V2-P1-S2-D9	221.583	124.207	78.40%
	5M-20J-V2-P1-S2-D10	219.214	109.607	100.00%
GROUP 31	5M-20J-V2-P2-S1-D1	148.98	107.833	38.16%
	5M-20J-V2-P2-S1-D2	216.866	107.381	101.96%
	5M-20J-V2-P2-S1-D3	206.465	122.874	68.03%
	5M-20J-V2-P2-S1-D4	198.166	107.689	84.02%
	5M-20J-V2-P2-S1-D5	155.272	119.121	30.35%
	5M-20J-V2-P2-S1-D6	242.388	102.968	135.40%
	5M-20J-V2-P2-S1-D7	152.526	101.016	50.99%
	5M-20J-V2-P2-S1-D8	188.395	120.088	56.88%
	5M-20J-V2-P2-S1-D9	195.063	118.777	64.23%
	5M-20J-V2-P2-S1-D10	159.774	117.481	36.00%
GROUP 32	5M-20J-V2-P2-S2-D1	211.831	113.212	87.11%
	5M-20J-V2-P2-S2-D2	190.783	108.007	76.64%
	5M-20J-V2-P2-S2-D3	253.579	122.477	107.04%
	5M-20J-V2-P2-S2-D4	169.478	107.689	57.38%
	5M-20J-V2-P2-S2-D5	194.394	117.793	65.03%
	5M-20J-V2-P2-S2-D6	145.041	103.868	39.64%
	5M-20J-V2-P2-S2-D7	157.816	103.719	52.16%
	5M-20J-V2-P2-S2-D8	206.776	121.496	70.19%
	5M-20J-V2-P2-S2-D9	176.904	117.212	50.93%
	5M-20J-V2-P2-S2-D10	185.059	115.522	60.19%