YAŞAR UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

MASTER THESIS

# BI-OBJECTIVE NO-WAIT PERMUTATION FLOWSHOP SCHEDULING PROBLEMS

DAMLA YÜKSEL

THESIS ADVISOR: PROF. DR. LEVENT KANDİLLER
CO-ADVISOR: PROF. DR. MEHMET FATİH TAŞGETİREN

INDUSTRIAL ENGINEERING

PRESENTATION DATE: 28.06.2019

BORNOVA / İZMİR
JUNE 2019

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:** |
|---|---|
| Prof. Dr. Levent KANDİLLER (Advisor)<br>Yaşar University | ..................... |
| Prof. Dr. Mehmet Cemali DİNÇER<br>Yaşar University | ..................... |
| Assoc. Prof. Dr. Özgür ÖZPEYNİRCİ<br>İzmir University of Economics | ..................... |

---------------------------------------------

Prof. Dr. Cüneyt GÜZELİŞ
Director of the Graduate School

# ABSTRACT

BI-OBJECTIVE NO-WAIT PERMUTATION FLOWSHOP SCHEDULING PROBLEMS

Yüksel, Damla
MSc, Industrial Engineering
Advisor: Levent KANDİLLER, PhD
Co-Advisor: Mehmet Fatih TAŞGETİREN, PhD
June 2019

In the field of permutation flowshop scheduling problems, there is a vast literature covering mathematical models and heuristics approaches. However, less work has been reported in the field of no-wait permutation flowshop scheduling problems, a variant of permutation flow shop scheduling problem where the waiting time for the jobs between the machines is not allowed. This thesis proposes both mixed-integer linear programming and constraint programming model formulations for no-wait permutation flowshop scheduling problem under various objectives such as (i) makespan, (ii) total flow time and (iii) total tardiness.

Moreover, energy-efficient scheduling has become very popular recently since energy consumption in high volume manufacturing is the leading essential difficulty in most industries. Both mixed-integer programming and constraint programming model formulations are developed in this thesis on the energy-efficient (bi-objective) no-wait permutation flowshop scheduling problems with the objective of minimizing (i) makespan, (ii) total flow time and (iii) total tardiness, separately. The bi-objective no-wait permutation flowshop scheduling problems treat the total energy consumption as a second objective in this study. Furthermore, due to the NP-hardness nature of the first objective of the problem, a novel multi-objective discrete artificial bee colony algorithm (MO-DABC), a traditional multi-objective genetic algorithm (MO-GA) and a variant of multi-objective genetic algorithm (MO-GALS) are proposed for the bi-objective no-wait permutation flowshop scheduling problems. Consequently, a comprehensive comparative metaheuristic analysis is carried out.

Hence, this thesis contributes to the literature of no-wait permutation flowshop scheduling problem for not only single-objective problems but also the bi-objective problems which consider energy efficient scheduling by ensuring various new mathematical models and metaheuristics.

**Key Words:** no-wait permutation flowshop scheduling problems, mixed-integer linear programming, constraint programming, bi-objective optimization, metaheuristics

# ÖZ

## İKİ AMAÇLI BEKLEMESİZ PERMUTASYON AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİ

Yüksel, Damla
Yüksek Lisans, Endüstri Mühendisliği
Danışman: Prof. Dr. Levent KANDİLLER
Yardımcı Danışman: Prof. Dr. Mehmet Fatih TAŞGETİREN
June 2019

Permütasyon akış tipi çizelgeleme problemlerinin literatür incelenmesinde matematiksel modellerin ve sezgisel yaklaşımların yaygın olarak kullanılmakta olduğu görülmüştür. Ancak, makineler arasındaki işler için bekleme süresine izin verilmeyen bir permütasyon akış tipi çizelgeleme problemi çeşidi olan beklemesiz permütasyon akış tipi çizelgeleme problemleri alanında daha az çalışma yapılmıştır. Bu tez, beklemesiz permütasyon akış tipi çizelgeleme problemi için hem karma-tamsayılı doğrusal programlama hem de kısıt programlama model formülasyonunu, (i) iş üretim süresi, (ii) toplam akış süresi ve (iii) toplam gecikme gibi çeşitli amaçlar altında önermektedir.

Ek olarak, enerji verimli çizelgeleme son zamanlarda oldukça popüler hale gelmiştir, çünkü yüksek hacimli imalattan kaynaklanan enerji tüketimi çoğu sektörde karşılaşılan en başta gelen problemdir. Bu nedenle, hem karma-tamsayılı programlama hem de kısıt programlama model formülasyonları, iki-amaçlı beklemesiz permütasyon akış tipi çizelgeleme problemleri üzerinde, yine (i) iş üretim süresini, (ii) toplam akış süresini ve (iii) toplam gecikmeyi ayrı ayrı en aza indirmek amacıyla çalışılmıştır. Bu tezde, iki-amaçlı beklemesiz permütasyon akış tipi çizelgeleme problemlerinin toplam enerji tüketimini ikinci bir amaç olarak kullandığı kabul edilmiştir. Ayrıca, ilk amaç fonksiyonunda bile NP-Hard sınıfında olan bu problem kapsamında, iki-amaçlı beklemesiz permütasyon akış tipi çizelgeleme problemleri için yeni bir çok-amaçlı ayrık yapay arı kolonisi algoritması (MO-DABC), bir geleneksel çok-amaçlı genetik algoritma (MO-GA) ve bir çok-amaçlı genetik algoritma çeşidi (MO-GALS) önerilmiştir. Sonuç olarak, iki amaçlı beklemesiz permütasyon akış tipi çizelgeleme problemleri için kapsamlı bir karşılaştırmalı metasezgisel analiz yapılmıştır.

Bu nedenle, bu tez, beklemesiz permütasyon akış tipi çizelgeleme problemi literatürüne sadece tek-hedefli problemler için değil aynı zamanda iki-hedefli enerji verimli çizelgeleme problemleri için çeşitli yeni matematiksel modeller ve metasezgisel yöntemler sağlayarak katkıda bulunmaktadır.

**Anahtar Kelimeler:** beklemesiz permütasyon akış tipi çizelgeleme problemleri, karma-tamsayılı lineer programlama, kısıt programlama, iki-amaçlı optimizasyon, sezgisel yöntemler

## ACKNOWLEDGEMENTS

First of all, I would like to express my special appreciation and sincere gratitude to my thesis advisor Prof. Dr. Levent Kandiller for his immense knowledge, guidance and patience during this study. His valuable guidance significantly helped in all the time of research and writing of this thesis. His father-like approach supported me in overcoming numerous obstacles and in moving in the right direction whenever I thought I needed it. I could not have imagined having a better advisor for my master study. Secondly, I would specially like to thank my thesis co-advisor Prof. Dr. Mehmet Fatih Taşgetiren for his substantial guidance during this way. He always shared his ideas with me to acquire results of better quality and to enhance my skills as a researcher. Also, I would like to thank Prof. Dr. Deniz Türsel Eliiyi for her help during the early stages of this thesis. Moreover, I appreciate to jury members Prof. Dr. Mehmet Cemali Dinçer and Assoc. Prof. Dr. Özgür Özpeynirci for their many insightful comments, suggestions and contributions.

Next, I would particularly like to thank my beloved husband Yiğit Yüksel for the support, love, as well as immutable encouragement I have gotten over the years. He always exchanged ideas and facilitated my improved skills and expand my point of view during each and every single step along this way. In particular, I would like to express my profound gratitude to my mother, my father and my sister Deniz. I undoubtedly could not have done this without their unfailing support and continuous encouragement.

Finally, the last but by no means the least, I have spent a great amount of time at the Department of Industrial Engineering in the Yaşar University and it was like my second home. Thus, I appreciate to faculty, colleagues and especially my dearest office mates whose endorsement are always with me.

Thanks for all your encouragement!

Damla Yüksel

İzmir, 2019

## TEXT OF OATH

I declare and honestly confirm that my study, titled "BI-OBJECTIVE NO-WAIT PERMUTATION FLOWSHOP SCHEDULING PROBLEMS" 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.

Damla Yüksel

Signature

........................................

July 2, 2019

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

## SYMBOLS AND ABBREVIATIONS

ABBREVIATIONS:

| | |
|---|---|
| PFSP | Permutation Flowshop Scheduling Problem |
| NWPFSP | No-Wait Permutation Flowshop Scheduling Problem |
| BI-OBJ NWPFSP | Bi-Objective No-Wait Permutation Flowshop Scheduling Problem |
| MILP | Mixed Integer Linear Programming |
| CP | Constraint Programming |
| EDD | Earliest Due Date |
| $F_m \| nwt \| C_{max}$ | No-Wait Permutation Flowshop Scheduling Problem for Minimization of Makespan |
| $F_m \| nwt \| \sum C_{iM}$ | No-Wait Permutation Flowshop Scheduling Problem for Minimization of Total Flow Time |
| $F_m \| nwt \| \sum T_i$ | No-Wait Permutation Flowshop Scheduling Problem for Minimization of Total Tardiness |
| $F_m \| nwt \| C_{max}, TEC$ | Bi-Objective No-Wait Permutation Flowshop Scheduling Problem for Minimization of Makespan and Total Energy Consumption |
| $F_m \| nwt \| \sum C_{iM}, TEC$ | Bi-Objective No-Wait Permutation Flowshop Scheduling Problem for Minimization of Total Flow Time and Total Energy Consumption |
| $F_m \| nwt \| \sum T_i, TEC$ | Bi-Objective No-Wait Permutation Flowshop Scheduling Problem for Minimization of Total Tardiness and Total Energy Consumption |
| $F_m \| nwt, d_i \| C_{max}$ | No-Wait Permutation Flowshop Scheduling Problem for Minimization of Makespan with Due Date Constraints |

| | |
|---|---|
| $F_m\|nwt, ST_{si}\|C_{max}$ | No-Wait Permutation Flowshop Scheduling Problem for Minimization of Makespan with Sequence Independent Setup Times |
| $F_m\|nwt, ST_{sd}\|C_{max}$ | No-Wait Permutation Flowshop Scheduling Problem for Minimization of Makespan with Sequence Dependent Setup Times |
| DC | Destruction and Construction |
| IGA | Iterated Greedy Algorithm |
| BIH | Block Insertion Heuristic |
| ILS | Insertion Local Search |
| MO-DABC | Multi-Objective Discrete Artificial Bee Colony Algorithm |
| MO-GA | Multi-Objective Generic Algorithm |
| MO-GALS | Multi-Objective Genetic Algorithm with a Local Search |

SYMBOLS:

| | |
|---|---|
| $N$ | Number of jobs |
| $M$ | Number of machines |
| $i, k$ | Indices for jobs $(1 \leq i, k \leq \|N\|)$ |
| $r$ | Index for machines $(1 \leq r \leq \|M\|)$ |
| $P_{ir}$ | Processing time of job $i \in N$ on machine $r \in M$ |
| $DD_i$ | Due date of job $i \in N$ |
| $Q$ | A very large number |
| $C_{ir}$ | Completion time of job $i \in N$ on machine $r \in M$ |
| $D_{ik}$ | 1 if job $i \in N$ is scheduled any time before job $k \in N$ , 0 otherwise $(i < k)$ |
| $T_i$ | Tardiness of job $i \in N$ |

| | |
|---|---|
| $C_{max}$ | Maximum completion time (makespan) |
| $Job_{ir}$ | Interval variable for the processing time ($P_{ir}$) of job $i \in N$ on machine $r \in M$ |
| $Mac_r$ | Sequence variable for machines over all $JobInt_{ir}$ |
| $s_l$ | Speed factor of processing speed level $l \in L$ |
| $\lambda_l$ | Conversion factor for processing speed level $l \in L$ |
| $\varphi_r$ | Conversion factor for idle time on machine $r \in M$ |
| $\tau_r$ | Power of machine $r \in M$ (kW) |
| $y_{irl}$ | 1 if job $i$ is processed at speed $l$ on machine $\in M$ , 0 otherwise |
| $\theta_r$ | Idle time on machine $r \in M$ |
| $TEC$ | Total energy consumption (kWh) |
| $JobOpt_{irl}$ | Optional interval variable for the processing time ($P_{ir}$) of job $i \in N$ on machine $r \in M$ which has the speed levels of $l \in L$ |
| $d[i,j]$ | Minimum distance between two consecutive jobs *i* and *j* on the first machine |
| $d([i,j][l,t])$ | Minimum distance between two consecutive jobs *i* and *j* on the first machine if their speed levels are *l* and *t*, respectively. |

# CHAPTER 1
# INTRODUCTION

The no-wait permutation flowshop scheduling problem (NWPFSP) is a variant of permutation flowshop scheduling problems (PFSP) that are most studied scheduling problems yielding significant practical applications in chemical, steel, plastic, food-processing, pharmaceutical and electronic industries (Aldowaisan & Allahverdi, 2004; Sapkal & Laha, 2013). The no-wait flowshop scheduling problems differ from the traditional flowshop problems by the following extra restriction: any holding up is not allowed between two consecutively used machines for any job. In the manufacturing areas of the above mentioned industries, owing to the technological restrictions, there might not be any storage area between two machines, which proceed the same job successively. In other words, once a job starts processing on the first machine, it must be processed without disruption until the end of the processing on the last machine. Furthermore, it is necessary to mention that the no-wait flowshop scheduling problems are perceived as permutation flowshop problems where each job follows the same order on machines (Fink & Voß, 2003). Hence, NWPFSPs target to find a job sequence for all jobs on the machines where each job follows the same order on the machines and where there is no waiting time between consecutive machines for a job.

This thesis aims to investigate a new fundamental mathematical modelling for three important NWPFSPs: (i) no-wait flowshop scheduling problem with the objective of makespan minimization, (ii) no-wait flowshop scheduling problem with the objective of total flow time minimization, and (iii) no-wait flowshop scheduling problem with the objective of total tardiness minimization. To be consistent with the standard 3-tuple notation framework, these problems are denoted as $(F_m|nwt|C_{max})$, $(F_m|nwt|\sum C_{iM})$ and $(F_m|nwt|\sum T_i)$, respectively (Graham et al. 1979). According to this notation, $F_m$ represents a flowshop with $m$ machines and $nwt$ indicates the no-wait restriction of the jobs between successive machines. $C_{max}$, $\sum C_{iM}$ and $\sum T_i$ denote that the objective is to minimize the makespan, the total flow time and the total tardiness, respectively.

Moreover, this thesis contributes to the energy efficient scheduling literature in such a way that the total energy consumption is considered. Recently, the energy efficient scheduling and energy consumption consideration increase its popularity within the scope of scheduling environment due to the fact that the high-energy consumption is the biggest current concern during production (Fang et al., 2011). To reduce the high-energy consumption in the manufacturing environment, energy-efficient machines are required to be used. On the other hand, this might not be applicable in most of the processes due to the substantial amount of financial investment. Therefore, energy-efficiency concept is employed at the operational planning level on machines by means of developing solution techniques for energy efficient no-wait permutation flowshop scheduling problems, in this thesis. Hence, this thesis aims to investigate a novel fundamental mathematical modelling for three important bi-objective NWPFSPs: (i) bi-objective no-wait flowshop scheduling problem to minimize the makespan and the total energy consumption, (ii) bi-objective no-wait flowshop scheduling problem to minimize the total flow time and the total energy consumption, and (iii) bi-objective no-wait flowshop scheduling problem to minimize the total tardiness and the total energy consumption. Again, to be consistent with the standard 3-tuple notation framework, these problems are denoted as $(F_m|nwt|C_{max}, TEC)$ and $(F_m|nwt|\sum C_{iM}, TEC)$, $(F_m|nwt|\sum T_i, TEC)$, respectively. Here, the term *TEC* denotes the total energy consumption. We propose two types of models formulating: mixed-integer linear programming (MILP) and constraint programming (CP). In addition, since the problem is NP-Hard, a multi-objective discrete artificial bee colony (MO-DABC), and a multi-objective genetic algorithm (MO-GA), also a variant of this algorithm (MOGALS) are developed, as heuristic solution methods. The performance of these algorithms over MILP and CP are measured in terms of small sized instances, then the comparative performance of the heuristic algorithms is measured in terms of larger instances with respect to various performance metrics.

The remainder of the thesis is organized as follows: In Chapter 2, the problem definition and an extensive literature review for single-objective NWPFSPs and energy efficient scheduling problems are presented. Then, the mathematical models and constraint programming models for single-objective NWPFSPs and bi-objective NWPFSPs are provided in Chapter 3 and Chapter 4, respectively. While three energy efficient metaheuristics are proposed for bi-objective NWPFSPs in Chapter 5, the computational results are provided in Chapter 6. Finally, the concluding remarks are stated and future research directions are addressed in Chapter 7.

# CHAPTER 2
# PROBLEM DEFINITION AND LITERATURE REVIEW

In this chapter, initially, the problem definitions are stated for single-objective NWPFSPs and bi-objective (energy-efficient) NWPFSPs. Then, all sets, parameters and decision variables are presented for both mixed-integer linear programming model (MILP) and constraint programming model (CP) by reflecting the insight of the no-wait restrictions. Next, a comprehensive literature review for both single-objective NWPFSPs and energy efficient scheduling problems are provided. At the end, the gaps found in the literature and the contribution of this thesis to the literature are discussed.

## 2.1. Single-Objective No-Wait Permutation Flowshop Scheduling Problems

The no-queue restriction makes the NWPFSP a special variant of PFSPs. Hence, the assumptions of the NWPFSPs can be summarized as follows.

- Each job can only be processed in one machine at a time and each machine can only process one job at a time.
- Each machine must process the jobs in an identical order, meaning that this problem is a variant of permutation flow shop scheduling problems.
- The jobs cannot wait between the machines. Once they start being processed, they must complete their processes until the last machine without any interruption between machines. This is referred as the no-wait restriction.
- Each job either follows a job or proceeds a job on all machines.

According to these assumptions, the Gantt chart for an example of NWPFSP, where 4 jobs is required to be scheduled on 3 machines, can be seen in Figure 2.1.

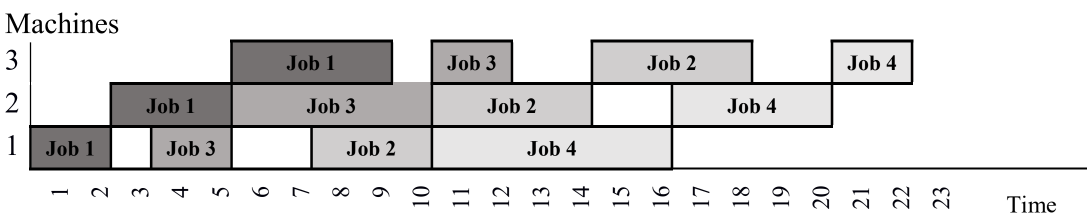


**Figure 2.1.** Gantt Chart for an Example of NWPFSP

The no-wait permutation flowshop scheduling problem (NWPFSP) is increasing its popularity due to the high level of applicability in most industries. This type of scheduling problems can be applicable when there is no any buffer spaces between the machines and/or when it is compulsory for the jobs to be processed without any interruption (Hall & Sriskandarajah, 1996). For example, in the food processing industry, the cooking operation must be started immediately after the canning operation for the assurance of freshness. Next, in silverware or plastic molding manufacturing industries, a bunch of operations required to be proceeded one after another. In chemical and pharmaceutical industries, the same reason of consecutive operations exists. Another example is steel production. Molding, unmolding, preliminary rolling, etc. require continuous production processes. Ultimately, no-wait scheduling problem arises even on the service industries where waiting in process of customers might inevitably costs high.

Various objective functions can be studied for scheduling problems such as makespan, maximum flow time, total flow time, total weighted flow time, maximum tardiness, maximum lateness, maximum earliness, total lateness, total tardiness, total earliness, total weighted tardiness, total number of tardy jobs, etc. However, in this thesis, three objective functions; (i) makespan, (ii) total flow time and (iii) total tardiness are studied separately. In other words, $(F_m|nwt|C_{max})$, $(F_m|nwt|\sum C_{iM})$ and $(F_m|nwt|\sum T_i)$ problems are employed. Hence, the sets, indices, parameters and decision variables are provided below:

| **Sets and Indices** | |
|---|---|
| $N$ | Set of jobs |
| $M$ | Set of machines |
| $i, k$ | Indices for jobs $(1 \leq i, k \leq \|N\|)$ |
| $r$ | Index for machines $(1 \leq r \leq \|M\|)$ |

| **Parameters** | |
|---|---|
| $P_{ir}$ | Processing time of job $i \in N$ on machine $r \in M$ |
| $DD_i$ | Due date of job $i \in N$ |
| $Q$ | A very large number |

| **Decision Variables for MILP Model** | |
|---|---|
| $C_{ir}$ | Completion time of job $i$ on machine $r$ |
| $D_{ik}$ | 1 if job $i$ is scheduled any time before job $k$, 0 otherwise $(i < k)$ |
| $T_i$ | Tardiness of job $i \in N$ |

| | |
|---|---|
| $C_{max}$ | Maximum completion time (makespan) |

| **Decision Variables for CP Model** | |
|---|---|
| $Job_{ir}$ | Interval variable for the processing time ($P_{ir}$) of job $i$ on machine $r$ |
| $Mac_r$ | Sequence variable for machines over all $JobInt_{ir}$ |

$N$ and $M$ stands for set of jobs and set of machines, respectively. Also, $i$ and $k$ are used for jobs indices, and $r$ is used for machine index. In this problem framework, the processing times $P_{ir}$ are preliminarily obtained so that the sequencing will depend on the processing times of jobs. Furthermore, due dates $DD_i$ of jobs are required for the $(F_m|nwt|\sum T_i)$ problem. $Q$ is provided for only the MILP model and it will be used in the sequencing constraints. $C_{ir}$ is the completion time of job $i$ on machine $r$. Namely, it equals to the starting time of job $i$ on the first machine plus its processing times on all machines, due to the no-wait restriction. $D_{ik}$ will be used to distinguish the jobs sequence in such a way that it takes the value of 1 if job $i$ is scheduled any time before job $k$, or 0 otherwise $(i < k)$. But this condition is checked for only the situations where $i < k$ since the construction of MILP model. For the $(F_m|nwt|\sum T_i)$ problem, the tardiness $T_i$ is calculated as follows: $T_i = \max\{(C_{iM} - DD_i),0\}$ means that the tardiness of each job $i$ equals to either the maximum of the difference between the completion time of the jobs on the last machine and its due date or 0. Namely, if the completion time of a job on the last machine goes beyond the due date of job, it becomes a tardy job. $C_{max}$ is the maximum completion time of all jobs, called as makespan. While considering the CP model, some additional decision variables are used due to the nature of CP. The interval variable $JobInt_{ir}$ converts the processing time information provided in the problem into a variable which keeps the information as an interval of a time during which job $i$ is processed on machine $r$. Then, the sequence variable $Mac_r$ is reserved for the sequence information of all the jobs. All the detailed information about the single-objective MILP and CP models is given in Chapter 3.

## 2.2. Bi-Objective (Energy-Efficient) No-Wait Permutation Flowshop Scheduling Problem

Energy consumption consideration for resource efficient manufacturing has been rapidly increasing recently. The main reason is that the energy consumption in high volume is the leading essential difficulty in most industries (Fang et al., 2011). Therefore, reduction of energy and power consumption in manufacturing carries huge importance while maintaining the same service levels. Thus, the studies in the scope of energy-efficient scheduling literature are increasing rapidly. In this thesis, the energy consumption is studied in the no-wait permutation flow shop setting (NWPFSPs). The importance of the total energy consumption comes from the usage of high, normal and slow speed levels. If a machine processes in high speed level, it decreases the jobs' process time and vice versa. Therefore, the existence of speed levels reveals that *"lower speed levels uses up less energy but raises process times"* and *"higher speed levels uses up more energy but reduces process times"*. Speed scaling strategy is a novel contribution of the Fang et al. (2011) that allows the machines operate at different speed levels when the different jobs are processed. Therefore, the tradeoff between the processing time and the total energy consumption is an existing fact and this leads to two main conflicting objectives. Generally, these problems can be notated as $(F_m|nwt|ProcessTime, TEC)$. However, in this study, three objective functions ((i) makespan, (ii) total flow time and (iii) total tardiness) are again studied for the bi-objective (energy-efficient) no-wait permutation flowshop scheduling problems, separately. In other words $(F_m|nwt|C_{max}, TEC)$ , $(F_m|nwt|\sum C_{iM}, TEC)$ and $(F_m|nwt|\sum T_i, TEC)$ problems are focused. Hence, the used sets, indices, parameters and decision variables are provided below:

| **Sets and Indices** | |
|---|---|
| $N$ | Set of jobs |
| $M$ | Set of machines |
| $L$ | Set of speed levels |
| $i$ and $k$ | Indices for jobs $(1 \leq i, k \leq |N|)$ |
| $r$ | Index for machines $(1 \leq r \leq |M|)$ |
| $l$ | Index for speed levels $(1 \leq l \leq |L|)$ |

| **Parameters** | |
|---|---|
| $P_{ir}$ | Processing time of job $i \in N$ on machine $r \in M$ |
| $DD_i$ | Due date of job $i \in N$ |
| $Q$ | A very large number |
| $s_l$ | Speed factor of processing speed level $l \in L$ |
| $\lambda_l$ | Conversion factor for processing speed level $l \in L$ |

| | |
|---|---|
| $\varphi_r$ | Conversion factor for idle time on machine $r \in M$ |
| $\tau_r$ | Power of machine $r \in M$ (kW) |

| **Decision Variables for MILP Model** | |
|---|---|
| $C_{ir}$ | Completion time of job $i$ on machines $r$ |
| $D_{ik}$ | 1 if job $i$ is scheduled any time before job $k$, 0 otherwise ($i < k$) |
| $T_i$ | Tardiness of job $i \in N$ |
| $C_{max}$ | Maximum completion time (makespan) |
| $y_{irl}$ | 1 if job $i$ is processed at speed $l$ on machine $r$, 0 otherwise. |
| $\theta_r$ | Idle time on machine r |
| $TEC$ | Total energy consumption (kWh) |

| **Decision Variables for CP Model** | |
|---|---|
| $Job_{ir}$ | Interval variable for the processing time ($P_{ir}$) of job $i$ on machine $r$ |
| $Mac_r$ | Sequence variable for machines over all $JobInt_{ir}$ |
| $JobOpt_{irl}$ | Optional interval variable for the processing time ($P_{ir}$) of job $i$ on machine $r$ which has the speed levels of $l$ |
| $\theta_r$ | Idle time on machine r |
| $C_{max}$ | Maximum completion time (makespan) |
| $TEC$ | Total energy consumption (kWh) |

Additively to the single-objective NWPFSPs, some other settings are needed to be made. For example, $L$ stands for set of speed levels. Because the speed scaling strategy will be used in the bi-objective MILP and CP model formulations. Also, $l$ is used for denoting the speed levels. Then, $s_l$ is the equivalent of the factor effect of speed level $l$. $\lambda_l$ and $\varphi_r$ are the conversion factors for processing speed level $l$ and for idle time on machine $r$, respectively. $\tau_r$ stands for the power of machine $r$. All these parameters are inspired from Mansouri et al. (2016). Other detailed information about the bi-objective MILP and CP model formulations is given in Chapter 4.

## 2.3. Literature Review on No-Wait Permutation Flowshop Scheduling Problems

This problem is a variant of the permutation flow shop scheduling problem (PFSP) where the jobs cannot wait between two consecutive machines. In other words, once a job starts it processing on the first machine, it must be processed on all downstream machines until completion without any interruption. Because of the technological restrictions, NWPFSPs are applicable in various industries such as chemical, electronics, plastics, metal, etc. (Sapkal & Laha, 2013) where the processing of each job must be continuous from the start to the end without any interruption. This problem is NP-hard for three or more machines (Röck, 1984). Deman & Baker (1974) proved that the mean flow time for NWPFSPs where there is no intermediate queue between the machines, can be solved with a branch and bound algorithm for smaller size of instances such as up to 12 jobs, while Wismer (1972) presented a branch and bound algorithm for the makespan criterion.

The total flow time criterion is studied by using a discrete harmony search algorithm by embedding a local search procedure (Gao et al. 2011). The mean flow time is minimized in a two-stage flexible no-wait flow shop problem (Shafaei et al. 2011). Next, a hybrid harmony search algorithm is studied with the help of a speed up method to reduce the running time requirement (Gao et al. 2012). A discrete differential evolution algorithm with the variable neighborhood descent algorithm is applied in NWPFSP (Tasgetiren et al. 2007). Another study proposes an improved iterated greedy algorithm that uses a tabu-based reconstruction strategy for the local minima search by Ding et al. (2015). Next, four composite and two constructive heuristics are proposed to minimize the total flow time in Gao et al. (2013). A hybrid particle swarm optimization algorithm regarding memetic algorithm, where a local search is hybridized, is computed on this problem by Akhshabi et al. (2014). Also, as a very comprehensive study, an discrete particle swarm optimization algorithm is developed for both makespan and total flow time depending on various speed-up methods for both the insert and swap neighborhood structures (Pan et al. 2008). Also, there is a recent study of Chaudhry et al. (2018) that proposes a genetic algorithm to minimize total flow time. Besides, Ying et al. (2016) proposed a self-adaptive ruin-and-recreate algorithm for the $(F_m|nwt|\sum C_{iM})$ problem and this algorithm improves almost more than half of the benchmark instances.

There are recent studies in literature regarding NWPFSPs. A very comprehensive literature review is presented in Lin & Ying (2016) and the makespan criterion is studied by converting the problem into an asymmetric travelling salesperson problem for the $(F_m|nwt|C_{max})$ problem. Then, the problem is solved optimally by two metaheuristics. In this study, the optimal results of makespan minimization are reported. More importantly, the study of Samarghandi & Behroozi

(2017) proposed a mixed integer linear programming, two quadratic mixed integer programming and two constraint programming model formulations for the $(F_m|nwt, d_i|C_{max})$ problems where due date restrictions are considered for the makespan minimization.

Moreover, a particle swarm optimization algorithm by Samarghandi (2015) is employed on the $(F_m|nwt, d_i|C_{max})$ problem which minimizes makespan with due date restrictions. Ying & Lin (2018) converted the $(F_m|nwt, ST_{si}|C_{max})$ and $(F_m|nwt, ST_{sd}|C_{max})$ problems into asymmetric travelling salesman problem and find the optimal solutions as they did in Lin & Ying (2016). Furthermore, although Aldowaisan & Allahverdi (2012) developed a simulated annealing and a genetic algorithm with the aid of dispatching rules for the $(F_m|nwt|\sum T_i)$ problem, they suggested the same algorithms for the $(F_m|nwt, ST_{si}|\sum T_i)$ problem in Aldowaisan & Allahverdi (2015), as well.

In addition, from the multi-objective perspective, Tavakkoli-Moghaddam et al. (2008) studied the weighted mean completion time and the weighted mean tardiness simultaneously for the NWPFSP with the help of an immune algorithm and they indicated the performance of the proposed algorithm over a multi-objective genetic algorithm. Another bi-criteria study is the study of Pan et al. (2009). They proposed a novel differential evolution algorithm to minimize the makespan and the maximum tardiness at the same time and show that the proposed algorithm performs superior than multi-objective genetic algorithm regarding the quality, efficiency, robustness and diversity level.

Finally, as a review paper, Nagano & Miyata (2016) provided a very extensive literature search for constructive heuristics under several criterion objectives and classify the heuristics as simple and composite by mentioning also the improvement heuristics, as well.

## 2.4. Literature Review on Energy Efficient Scheduling

The companies are searching more energy efficient scheduling techniques since the high-energy consumption is the most current difficulty of industries (Fang et al., 2011). Thus, the studies within the scope of energy-efficient scheduling literature are kept increasing. The energy efficient scheduling approaches which consider improving energy efficiency are analyzed as a review study in Gahm et al. (2016). An operational method was revealed to minimize the energy consumption of manufacturing equipment in Mouzon et al. (2007). A novel method was proposed which indicates that an adequate amount of energy can be saved, if the machines are turned-off during their idle times. Then, this method was used for the single machine scheduling problem by the total tardiness and the total energy consumption minimization in (Mouzon & Yıldırım, 2008) and for flexible flow shop problem by the makespan and the total energy consumption minimization in Dai et al. (2013). Also, the single machine total tardiness problem with sequence dependent setup times is recently studied on energy efficiency scheduling framework (Taşgetiren et al. 2018). In scope of PFSP, job-based speed scaling strategy is used for two-machine sequence-dependent PFSP where both the makespan and the total energy consumption are minimized (Mansouri et al. 2016). An energy-efficient formulation is recently made where the total flow time and the total energy consumption are studied simultaneously in Öztop et al. (2018). Furthermore, a backtracking algorithm is proposed for the energy efficient PFSP (Lu et al. 2017). On the other hand, studies regarding environmental effects and/or energy consumption consideration which are employed on NWPFSPs have not been rather exist.

## 2.5. Discussion

To the best of our knowledge, there is only one article which proposes MILP and CP models for $(F_m|nwt, d_i|C_{max})$ problem that considers the due date restriction (Samarghandi and Behroozi, 2017) and few articles proposing MILP models for $(F_m|nwt|C_{max})$. However, there is no any study construct MILP or CP models for $(F_m|nwt|\sum C_{iM})$ and $(F_m|nwt|\sum T_i)$ problems. This thesis proposes MILP and CP model formulations for all $(F_m|nwt|C_{max})$, $(F_m|nwt|\sum C_{iM})$ and $(F_m|nwt|\sum T_i)$ problems to fill this gap in the literature. In addition, a number of valid inequalities have been studied for $(F_m|nwt|C_{max})$ problem as represented in Table 2.1.

**Table 2.1.** The Contribution of the Thesis to the Literature of NWPFSPs

| | $(F_m\|nwt\|C_{max})$ | $(F_m\|nwt\|\sum C_{iM})$ | $(F_m\|nwt\|\sum T_i)$ |
|---|---|---|---|
| **MILP** | √ | √ | √ |
| **CP** | √ | √ | √ |
| **Valid Inequalities** | √ | | |

Furthermore, there is no any study, to the best of our knowledge, on bi-objective (energy-efficient) NWPFSPs which aims to minimize the total energy consumption. Hence, a bi-objective MILP and CP model formulations have been proposed to fill that gap, where the jobs can be processed at different speed levels corresponding to different energy consumption levels. Moreover, due to the NP-Hardness of the problem, we develop three metaheuristics for bi-objective $(F_m|nwt|C_{max}, TEC)$, $(F_m|nwt|\sum C_{iM}, TEC)$ and $(F_m|nwt|\sum T_i, TEC)$ problems as depicted in Table 2.2.

**Table 2.2.** The Contribution of the Thesis to the Literature of Bi-Objective (Energy-Efficient) NWPFSPs

| | $(F_m|nwt|C_{max}, TEC)$ | $(F_m|nwt|\sum C_{iM}, TEC)$ | $(F_m|nwt|\sum T_i, TEC)$ |
|---|---|---|---|
| **MILP** | √ | √ | √ |
| **CP** | √ | √ | √ |
| **Meta-heuristics** | √ | √ | √ |

One last contribution is that, the small size instance generation scheme is proposed for the $(F_m|nwt|\sum T_i)$ problem, since there is no any small sized instances exist for this criterion objective.

# CHAPTER 3
# SINGLE OBJECTIVE NO-WAIT PERMUTATION FLOWSHOP SCHEDULING PROBLEMS

In this chapter, no-wait permutation flowshop scheduling problem (NWPFSP) with the objective of minimizing the makespan, the total flow time and the total tardiness have studied in Sections 3.1, 3.2, 3.3 respectively. Namely, $(F_m|nwt|C_{max})$, $(F_m|nwt|\sum C_{iM})$ and $(F_m|nwt|\sum T_i)$ problems are employed. Both Mixed Integer Linear Programming (MILP) and Constraint Programming (CP) models have developed for each objective, as presented in this chapter. In addition, the comparison of the models are represented at the end of each section.

## 3.1. No-Wait Permutation Flowshop Scheduling Problem with Minimizing Makespan

NWPFSP with the objective of minimizing makespan $(F_m|nwt|C_{max})$ aims to find such a sequence of jobs, which minimizes makespan, on machines by not allowing the jobs to wait between the machines. $(F_m|nwt|C_{max})$ problem with $m$ machines when m is greater than or equal to 4 is NP-Hard, whereas in polynomial time the $(F_m|nwt|C_{max})$ problem can be solved when $m$ equals to 2 (Röck, 1984). The proposed MILP and CP models for $(F_m|nwt|C_{max})$ are presented below.

### 3.1.1. Mixed Integer Linear Programming Model

The mixed integer linear programming model for the single objective no-wait permutation flow shop problem for minimization of the makespan $(F_m|nwt|C_{max})$ is given below:

| **Model 1.** The MILP Model for the $(F_m|nwt|C_{max})$ | | |
|---|---|---|
| **Objective** | | |
| Minimize $C_{max}$ | | (1-01) |
| **Constraints** | | |
| $C_{i1} \geq P_{1i}$ | $\forall i \in N$ | (1-02) |
| $C_{ir} - C_{i,r-1} \geq P_{ir}$ | $\forall i \in N, \forall r \in M: r \geq 2$ | (1-03) |
| $C_{ir} - C_{kr} + QD_{ik} \geq P_{ir}$ | $\forall i \in N : k > i, \forall r \in M$ | (1-04) |
| $C_{ir} - C_{kr} + QD_{ik} \leq Q - P_{kr}$ | $\forall i \in N : k > i, \forall r \in M$ | (1-05) |

| | | |
|---|---|---|
| $C_{max} \geq C_{iM}$ | $\forall i \in N$ | (1-06) |
| $C_{ir} - C_{i,r-1} \leq P_{ir}$ | $\forall i \in N, \forall r \in M: r \geq 2$ | (1-07) |
| $C_{ir} \geq 0$ | $\forall i \in N, \forall r \in M$ | (1-08) |
| $D_{ik} \in \{0,1\}$ | $\forall i, k \in N : k > i$ | (1-09) |

The objective function (1-01) minimizes the makespan. Constraint (1-02) allows that the completion time of the jobs is to be at least its processing time on the first machine. It is assured that the completion time of each job on machine $r$ can only be greater than or equal to the completion time of the job on machine $r-1$ plus to the processing time of the job on machine $r$ by constraint (1-03). Then, constraint sets (1-04) and (1-05) provide that job $k$ either follows the job $i$, or precedes the job $i$, in the sequence. Later, the makespan is set to the completion time of maximum of all the last job on the last machine by constraint (1-06). Next, constraint (1-07) assures that the completion time of each job on machine $r$ can only be less than or equal to the completion time of the job on machine $r-1$ plus to the processing time of the job on machine *r*. Hence, this constraint provides that the no-wait constraint together with the constraint (1-03). In other words, the differences between the completion time of each job on machine $r$ and the completion time of the job on machine $r-1$ must be equal to the processing time of the job on machine $r$. Lastly, the sign restrictions are given in (1-08) and (1-09). Note that this model formulation is an extension of the permutation flow shop problem of Manne (1960) with the addition of no-wait restriction. This model provides a basis for the thesis in such a way that all the following mathematical model formulations will be conducted by modifying required changes over this model.

### 3.1.2. Constraint Programming Model

The constraint programming model for the single objective no-wait permutation flow shop problem for minimization of makespan ($F_m|nwt|C_{max}$) is given below:

**Model 2.** The CP Model for the ($F_m|nwt|C_{max}$)

**Objective**

| | | |
|---|---|---|
| Minimize $max_{i \in N}(\text{ENDOF}(Job_{iM})$ | | (2-01) |
| **Constraints** | | |
| $\text{ENDATSTART}(Job_{ir}, Job_{i,r+1})$ | $\forall i \in N, \forall r \in M: r < M$ | (2-02) |
| $\text{NOOVERLAP}(Mac_r)$ | $\forall r \in M$ | (2-03) |

| | | |
|---|---|---|
| SAMESEQUENCE($Mac_1, Mac_r$) | $\forall r \in M : 1 < r$ | (2-04) |

The objective function (2-01) minimizes the maximum of the end of the job intervals on the last machines, that is makespan. Constraint (2-02) is basically the no-wait constraint which provides that the job interval of any given job $i$ on the machine $r$ will be end at the starting time of the job interval of the same job $i$ on the machine $r + 1$. Constraint (2-03) assures that there cannot be any overlap on the machines which means that each machine can only process one job at a time. Lastly, the same sequence for the jobs on each machine is preserved by constraint (2-04).

This proposed constraint programming model is an original model which contributes to the literature of the no-wait flow shop scheduling problems. As it was mentioned in Section 2.1 previously, there are few constraint programming models for $(F_m|nwt, d_i|C_{max})$ in the literature but they are looking to the problem from another point of view (Samarghandi & Behroozi, 2017). Hence, this model allows us to investigate various sides of the $(F_m|nwt|C_{max})$ problem.

### 3.1.3. Valid Inequalities for $(F_m|nwt|C_{max})$ Problem

For the $(F_m|nwt|C_{max})$ problem, four valid are proposed as represented below.

| Valid Inequalities | |
|---|---|
| $C_{max} \leq \sum_{i=1}^{\lvert N\rvert} \sum_{r=1}^{\lvert M\rvert} P_{i,r}$ | (V-01) |
| $C_{max} \leq \sum_{i=1}^{\lvert N\rvert} C_{iM}$ | (V-02) |
| $C_{max} \geq \sum_{i=1}^{\lvert N\rvert} P_{i,M}$ | (V-03) |
| $C_{max} \geq \sum_{i=1}^{\lvert N\rvert} P_{i,M} + \sum_{i=1}^{\lvert N\rvert} \sum_{r=1}^{\lvert M-1\rvert} y_i * P_{ir}$<br>$\sum_{i=1}^{\lvert N\rvert} y_i = 1 \ \forall_i$<br>$C_{i1} - P_{i1} \leq Q(1 - y_i) \ \forall_i$ | (V-04) |

(V-01) is an upper bound for the makespan that calculates the sum of processing time of all jobs on all machines. (V-02) is another upper bound for makespan that restricts the makespan as the sum of all completion time of all jobs. (V-03) is a lower bound which the sum of jobs' processing times on the last machine restricts the makespan. Lastly, the (V-04) is another lower bound that aims to find the first job in the sequence. If the job i is in the first position, then $y_i$ becomes 1. Thus, this bound sums the processing time of the first job and the processing time of other jobs on the last machine.

### 3.1.4. Computational Results and Comparison of the MILP and CP Models

Initially, the proposed MILP and CP models are run on the first part of the instances of Vallada et al. (2015), which consist of 240 small instances including 24 different combinations of $n = \{10,20,30,40,50,60\}$ jobs with $m = \{5,10,15,20\}$ machines. (These instances are also called as VRF instances, in the literature.) There are 10 instances of each combination. The optimal value (Lin & Ying, 2016), objective function of MILP, gap of MILP, objective function of CP model and gap of CP models are reported in Appendix A. The tables are prepared regarding the number of machines therefore the set of instances of *10,20,30,40,50,60 jobs x 5 machines*, *10,20,30,40,50,60 jobs x 10 machines, 10,20,30,40,50,60 jobs x 15 machines* and *10,20,30,40,50,60 jobs x 20 machines* are reported in Tables A.1., A.2., A.3. and A.4., respectively. Then, to be able to analyze the performance of the MILP and CP models, the averages of each set are calculated.

Next, the proposed valid inequalities are made use of and they are added to the MILP and CP models. The MILP model is named as MILP-Prime and the CP model is named as CP-Prime with the addition of all valid inequalities. The results are presented in Appendix A ( Table A.5, Table A.6., Table A.7., Table A.8.). (V-01) that is an upper bound and (V-03) that is a lower bound are calculated separately and reported in the mentioned tables. According to these results, the valid inequalities improves the performance of MILP model so that MILP-Prime performs better than MILP. However, the performance of CP-Prime is very close to CP model. Moreover, although the MILP-Prime is superior than MILP, its performance cannot exceed the performance of CP model. Finally, also the valid inequality (V-03) can be extended by addition of *minimum of sum of the processing times of each job until the last machine*.

The detailed results are provided in the following. The proposed MILP and CP come up with the optimal solutions for the instances with 10 jobs and 5,10,15,20 machines. The average results for these set of instances are given in Figure 3.1.

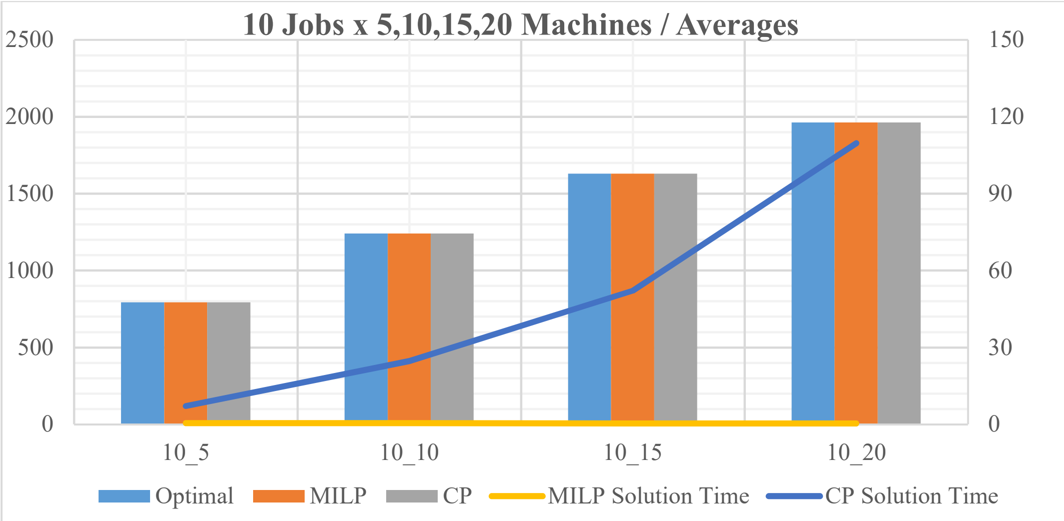


**Figure 3.1.** Comparison of MILP and CP models for $(F_m|nwt|C_{max})$ Problem on VRF Instances (Small Sized)

As seen in the Figure 3.1., the MILP and CP find the optimal solution for $(F_m|nwt|C_{max})$ problem. However, in terms of the computational time, MILP model performs better than the CP model on these small instances of 10 jobs. Then, the averages for the 5, 10, 15 and 20 machines are plotted on Figures 3.2., 3.3., 3.4., and 3.5., respectively based on the information obtained from the Tables A1.1., to A1.4.

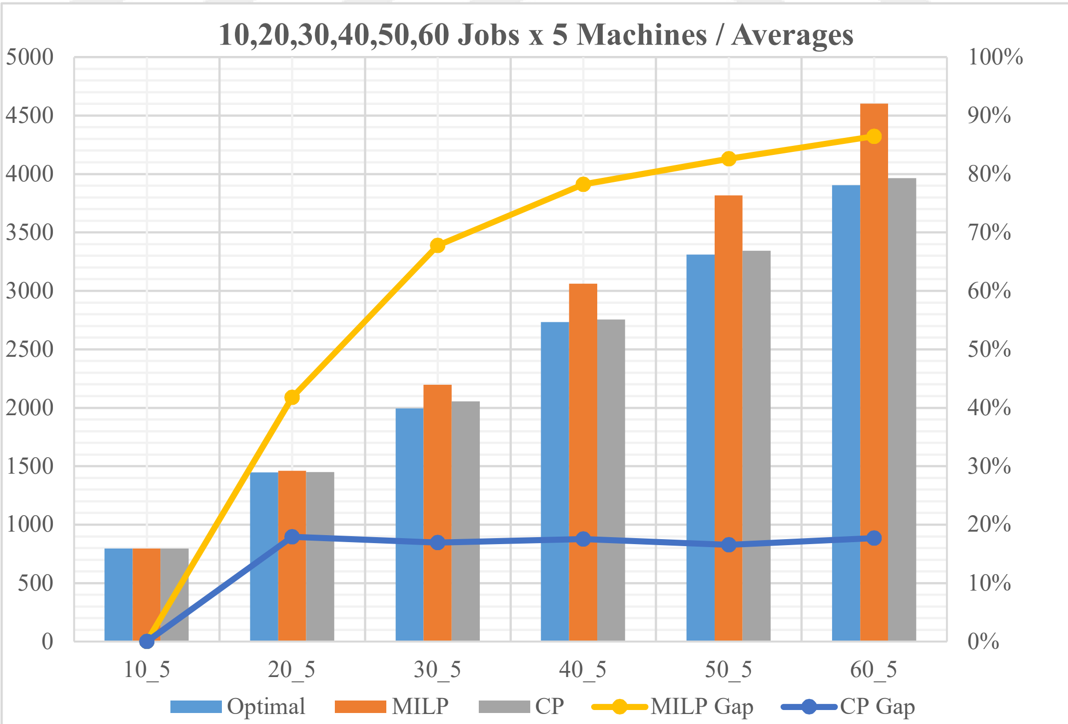


**Figure 3.2.** Comparison of MILP and CP Models for $(F_m|nwt|C_{max})$ Problem on VRF Instances (5 Machines)

According to the Figure 3.2., the average gap of CP model stays between 17% and 18% even though the number of jobs is increasing. However, the average gap of the MILP model increases significantly when the number of jobs is increasing. The maximum MILP gap is 86% while the CP gap is significantly less.

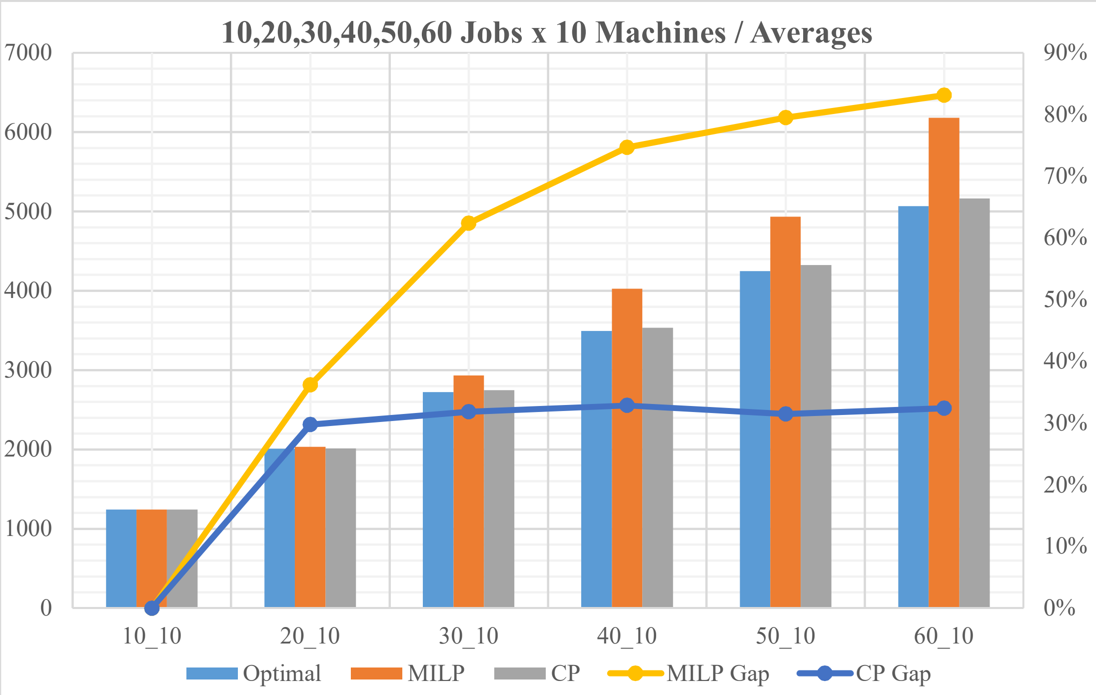


**Figure 3.3.** Comparison of MILP and CP Models for $(F_m|nwt|C_{max})$ Problem on VRF Instances (10 Machines)

According to the Figure 3.3., the average gap of CP model stays between 30% and 33% even though the number of jobs is increasing. However, the average gap of the MILP model increases significantly when the number of jobs is increasing. The maximum MILP gap is 83% while the CP gap is comparatively less.

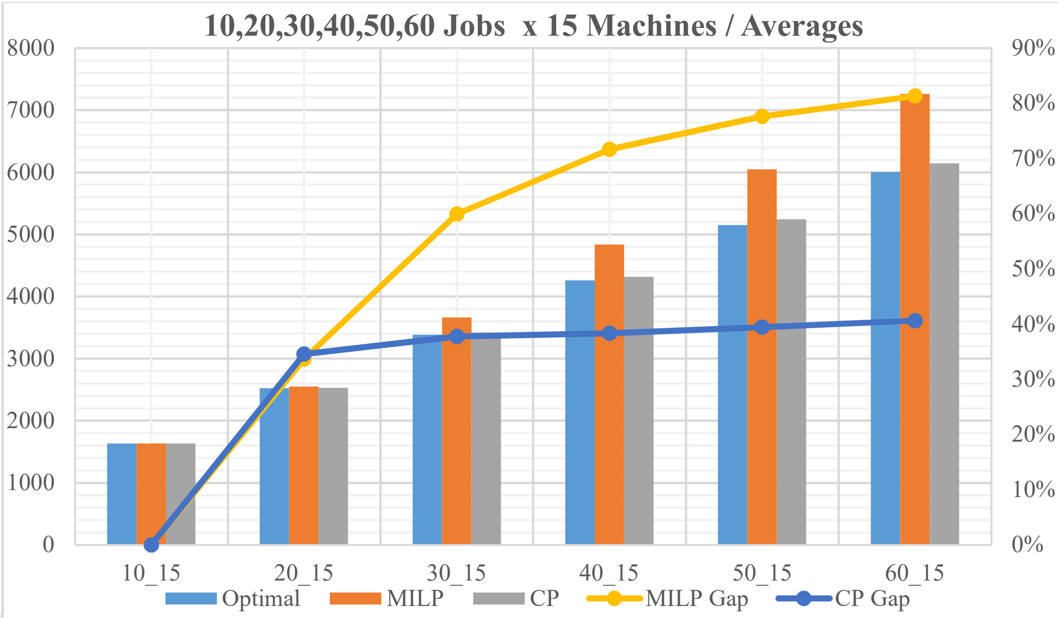


**Figure 3.4.** Comparison of MILP and CP Models for ($F_m|nwt|C_{max}$) Problem on VRF Instances (15 Machines)

According to the Figure 3.4., the average gap of CP model stays between 35% and 41% even though the number of jobs is increasing. However, the average gap of the MILP model increases significantly when the number of jobs is increasing. The maximum MILP gap is 81% while the CP gap is comparatively less.

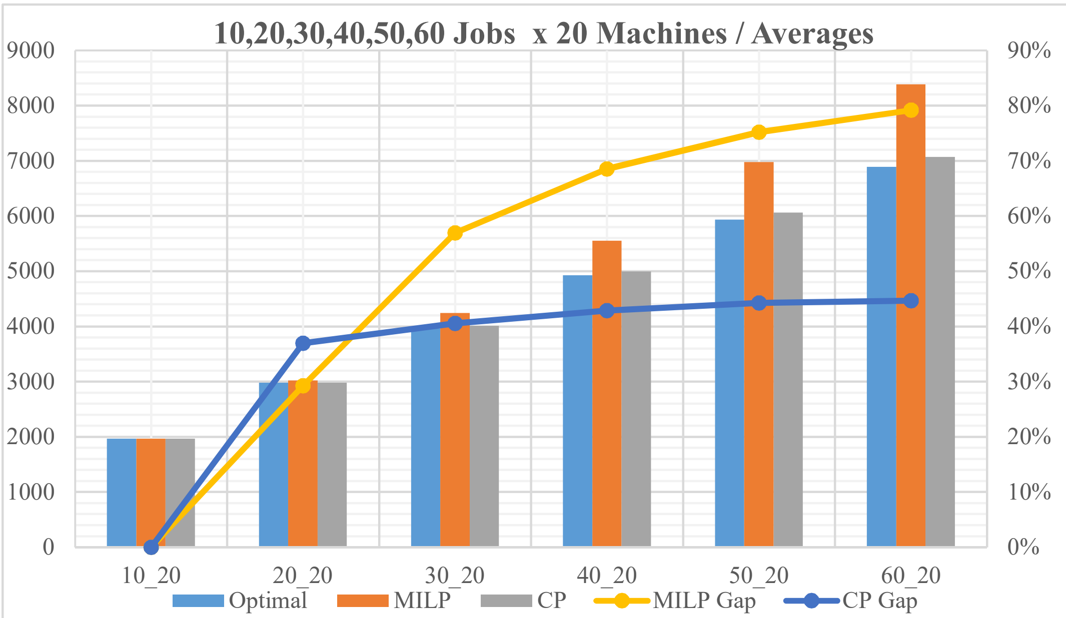


**Figure 3.5.** Comparison of MILP and CP Models for ($F_m|nwt|C_{max}$) Problem on VRF Instances (20 Machines)

According to the Figure 3.5., the average gap of CP model stays between 37% and 45% even though the number of jobs is increasing. However, the average gap of the MILP model increases significantly when the number of jobs is increasing. The maximum MILP gap is 79% while the CP gap is much less than the MILP gap.

To sum up, all of the graphs state that the gap of the CP model is not being affected by the changes in the number of jobs. Its trend follows a similar way when the number of jobs is increasing. In addition, CP model performs much better than MILP model in terms of the magnitude of the objective function and the gap percentage found in 3600 seconds.

From Appendix A (Tables A.1., A.2., A.3. and A.4.) and Appendix A (Tables A.5., A.6., A.7. and A.8.) MILP and MILP-Prime results can be seen, respectively. However, to see the effect of valid inequalities, a comparison table is presented in Table 3.1. According to Table 3.1., the benefits of valid inequalities can be interpreted, since the MILP-Prime model formulation finds almost the same objective function but with a lower percentage gap.

**Table 3.1.** Comparison of MILP and MILP-Prime in terms of Averages

| Instance | MILP | Time (Seconds) | Gap % | MILP-Prime | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|
| 10_5_Average | 793.60 | 0.427 | 0.00% | 793.60 | 0.628 | 0.00% |
| 10_10_Average | 1241.10 | 0.477 | 0.00% | 1241.10 | 1.002 | 0.00% |
| 10_15_Average | 1629.50 | 0.380 | 0.00% | 1629.50 | 1.730 | 0.00% |
| 10_20_Average | 1962.00 | 0.380 | 0.00% | 1962.00 | 1.875 | 0.00% |
| 20_5_Average | 1460.30 | 3600 | 41.77% | 1466.70 | 3600 | 27.42% |
| 20_10_Average | 2031.50 | 3600 | 36.20% | 2027.80 | 3600 | 37.03% |
| 20_15_Average | 2546.90 | 3600 | 33.66% | 2536.10 | 3600 | 35.44% |
| 20_20_Average | 3018.50 | 3600 | 29.25% | 2998.40 | 3600 | 32.42% |
| 30_5_Average | 2195.50 | 3600 | 67.77% | 2225.50 | 3600 | 31.57% |
| 30_10_Average | 2932.50 | 3600 | 62.39% | 2946.60 | 3600 | 43.12% |
| 30_15_Average | 3661.60 | 3600 | 59.91% | 3670.90 | 3600 | 49.33% |
| 30_20_Average | 4241.50 | 3600 | 56.95% | 4282.40 | 3600 | 50.99% |
| 40_5_Average | 3059.30 | 3600 | 78.24% | 3069.00 | 3600 | 32.81% |
| 40_10_Average | 4022.30 | 3600 | 74.67% | 4040.20 | 3600 | 46.19% |
| 40_15_Average | 4831.80 | 3600 | 71.63% | 4844.70 | 3600 | 50.81% |
| 40_20_Average | 5548.10 | 3600 | 68.55% | 5681.40 | 3600 | 55.96% |
| 50_5_Average | 3816.70 | 3600 | 82.60% | 3869.20 | 3600 | 34.26% |
| 50_10_Average | 4933.40 | 3600 | 79.48% | 5062.10 | 3600 | 46.97% |
| 50_15_Average | 6041.40 | 3600 | 77.63% | 6167.10 | 3600 | 53.27% |
| 50_20_Average | 6975.00 | 3600 | 75.20% | 7178.90 | 3600 | 57.45% |
| 60_5_Average | 4599.80 | 3600 | 86.45% | 4667.90 | 3600 | 35.03% |
| 60_10_Average | 6176.50 | 3600 | 83.12% | 6198.40 | 3600 | 47.75% |
| 60_15_Average | 7258.00 | 3600 | 81.29% | 7375.40 | 3600 | 53.91% |
| 60_20_Average | 8382.50 | 3600 | 79.16% | 8481.70 | 3600 | 57.92% |

Lastly, the proposed MILP and CP models are run 3600 seconds on the instances of Taillard (1993) with 11 different combinations of *20x5, 20x10, 20x20, 50x5, 50x10, 50x20, 100x5, 100x10, 100x20, 200x10* and *200x20* set of instances where the first number indicates the number of jobs and the second indicates the number of machines. There are 10 instances of each combination so that 110 instances are studied. All the objective function results are reported in Appendix A (Table A.9.,

Table A.10., Table A.11, Table A.12.). According to these results, similar with results of VRF instances, CP model performs superior than MILP model. Namely, the gaps of CP are lower than the gaps of MILP, also the objective function values of CP are lower than the objective function of MILP. Another important point here is that while MILP model cannot find any feasible solution for *100x10, 100x20, 200x10* and *200x20* instances, CP model can find a solution for those instances within 3600 seconds.

## 3.2. No-Wait Permutation Flowshop Scheduling Problem with Minimizing Total Flow Time

NWPFSP with the objective of minimizing the total flow time $(F_m|nwt|\sum C_{iM})$ aims to find the sequence of jobs which minimizes the total completion time of all jobs without permitting process queue for the jobs between the machines. The completion time of jobs can obviously be expressed as the sum of starting time of the job on the first machine and its total processing time on all machines.

### 3.2.1. Mixed Integer Linear Programming Model

The mixed integer linear programming model formulation for the single objective no-wait permutation flow shop problem for minimization of the total flow time $(F_m|nwt|\sum C_{iM})$ is given below:

| **Model 3.** The MILP Model for the $(F_m|nwt|\sum C_{iM})$ | | |
|---|---|---|
| **Objective** | | |
| Minimize $\sum_{i \in N} C_{iM}$ | | (3-01) |
| **Constraints** | | |
| $C_{i1} \geq P_{i1}$ | $\forall i \in N$ | (3-02) |
| $C_{ir} - C_{i,r-1} \geq P_{ir}$ | $\forall i \in N, \forall r \in M: r \geq 2$ | (3-03) |
| $C_{ir} - C_{kr} + QD_{ik} \geq P_{ir}$ | $\forall i \in N : k > i, \forall r \in M$ | (3-04) |
| $C_{ir} - C_{kr} + QD_{ik} \leq Q - P_{kr}$ | $\forall i \in N : k > i, \forall r \in M$ | (3-05) |
| $C_{ir} - C_{i,r-1} \leq P_{ir}$ | $\forall i \in N, \forall r \in M: r \geq 2$ | (3-06) |
| $C_{ir} \geq 0$ | $\forall i \in N, \forall r \in M$ | (3-07) |
| $D_{ik} \in \{0,1\}$ | $\forall i, k \in N : k > i$ | (3-08) |

The objective function (3-01) minimizes the total flow time. Constraint (3-02) provides that the completion time of the jobs is to be at least its processing time on the first machine. Constraint (3-03) assures that the completion time of each job on machine $r$ can only be greater than or equal to the completion time of the job on machine $r-1$ plus to the processing time of the job on machine $r$. Then, constraint (3-04) and (3-05) provide that job $k$ either follows the job $i$, or precedes the job $i$, but not both in the sequence. Next, it is provided that the completion time of each job on machine $r$ can only be less than or equal to the completion time of the job on machine $r-1$ plus to the processing time of the job on machine $r$ by constraint (3-06). Hence, this constraint satisfies the no-wait requirement together with the constraint (3-03). In other words, the differences between the completion time of each job on machine $r$ and the completion time of the job on machine $r-1$ must be equal to the processing time of the job on machine $r$. Lastly, the sign restrictions and binary variables are given in (3-07) and (3-08). Note that this model is an extension of the permutation flow shop problem of Manne (1960) by adding the no-wait constraint and by considering the total flow time in the objective function, too.

### 3.2.2. Constraint Programming Model

The constraint programming model for the single objective no-wait permutation flow shop problem for minimization of total flow time ($F_m|nwt|\sum C_{iM}$) is given below:

| **Model 4.** The CP Model for the ($F_m|nwt|\sum C_{iM}$) | | |
|---|---|---|
| **Objective** | | |
| Minimize $\sum_{i\in N}$ ENDOF($Job_{iM}$) | | (4-01) |
| **Constraints** | | |
| ENDATSTART($Job_{ir}$, $Job_{i,r+1}$) | $\forall i \in N, \forall r \in M: r < M$ | (4-02) |
| NOOVERLAP($Mac_r$) | $\forall r \in M$ | (4-03) |
| SAMESEQUENCE($Mac_1, Mac_r$) | $\forall r \in M : 1 < r$ | (4-04) |

The objective function (4-01) minimizes the sum of the end of the job intervals on the last machine, which is the total flow time. Constraint (4-02) is the no-wait constraint which provides that the job interval of any given job $i$ on the machine $r$ will be ended at the starting time of the job interval of the same job $i$ on the machine $r+1$. Constraint (4-03) provides that there cannot be any overlap on the machines which means that each machine can only process one job at a time. Lastly, the same sequence for the jobs on each machine is preserved by the constraint (4-04).

This proposed constraint programming model is an original model which adds value to the literature of the no-wait flow shop scheduling problems which minimize the total flow time.

### 3.2.3. Computational Results and Comparison of the MILP and CP Models

Initially, the proposed MILP and CP models are run on the first part of instances of Vallada et al. (2015) which consist of 240 small instances including 24 different combinations of $n = \{10,20,30,40,50,60\}$ jobs with $m = \{5,10,15,20\}$ machines. There are 10 instances of each combination. The objective function of MILP, gap of MILP, objective function of CP model and gap of CP models are reported in Appendix B. The tables are prepared regarding the number of machines; therefore, the set of instances of *10,20,30,40,50,60 jobs x 5 machines*, *10,20,30,40,50,60 jobs x 10 machines, 10,20,30,40,50,60 jobs x 15 machines* and *10,20,30,40,50,60 jobs x 20 machines* are reported in Tables B.1., to B.4., respectively. Then, to be able to analyze the performance of the MILP and CP models, the averages of each set are calculated. The proposed MILP and CP find the optimal solutions for the instances with 10 jobs and 5, 10, 15, 20 machines. The results for these instances are given in Figure 3.6.

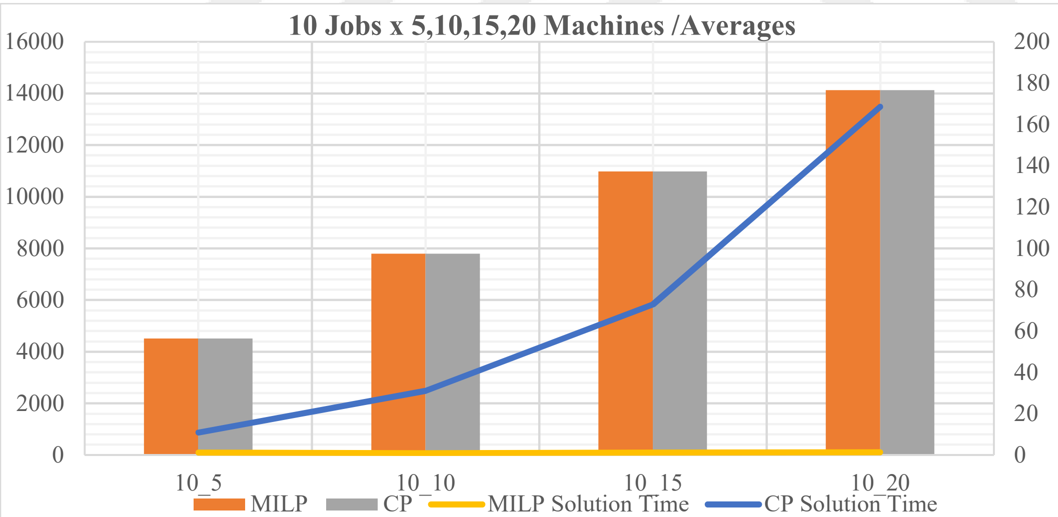


**Figure 3.6.** Comparison of MILP and CP models for $(F_m|nwt|\sum C_{iM})$ Problem on VRF Instances (Small Sized)

Figure 3.6. indicates that the MILP and CP models reach the optimal solution for $(F_m|nwt|\sum C_{iM})$ problem in a small run time. On the other hand, if models are compared with each other, the MILP model is better than the CP model in terms of the computational time.

Then, the averages for the 5, 10, 15 and 20 machines are plotted on the Figures 3.7. to 3.10., respectively based on the information obtained from the Tables B.1., to B.4.

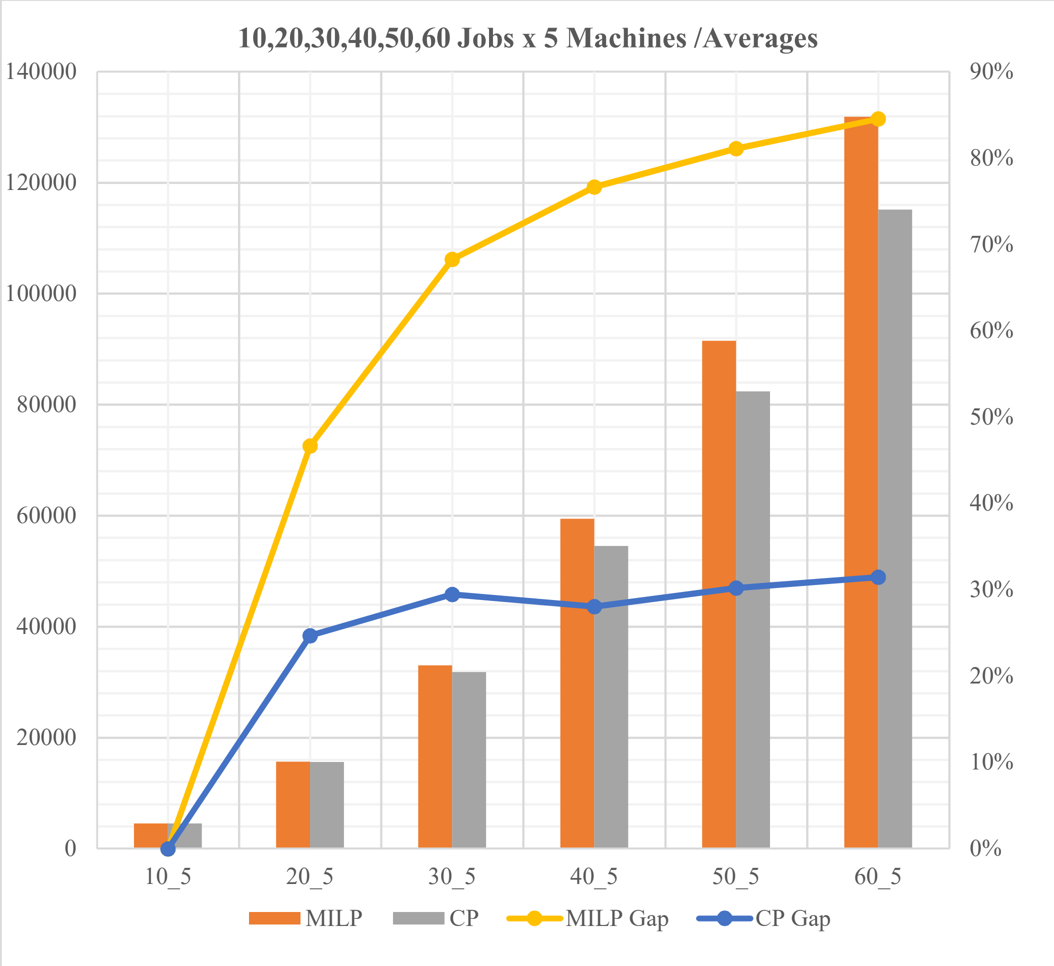


**Figure 3.7.** Comparison of MILP and CP Models for $(F_m|nwt|\sum C_{iM})$ Problem on VRF Instances (5 Machines)

According to the Figure 3.7., the average gap of the CP model stays between 25% and 31% even though the number of jobs is increasing. However, the average gap of the MILP model increases significantly when the number of jobs is kept increasing.

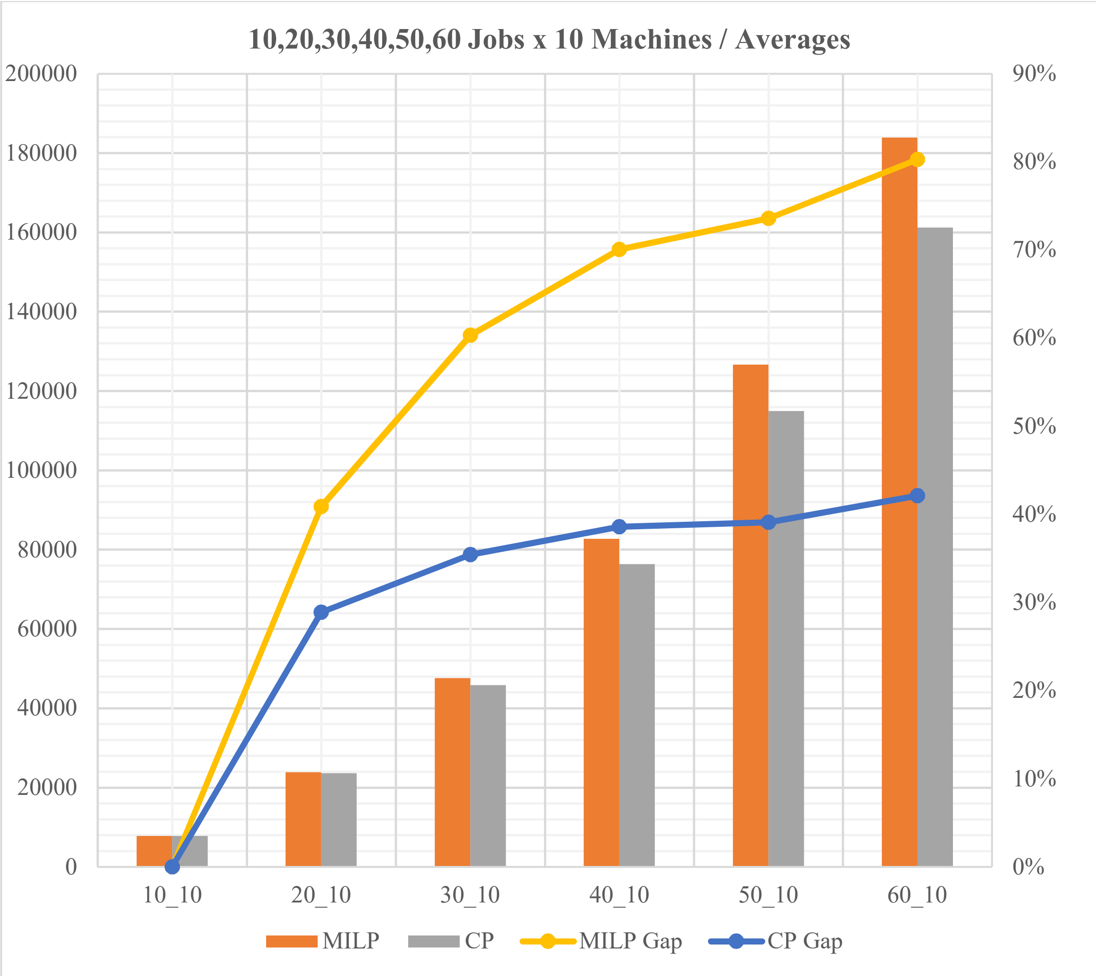


**Figure 3.8.** Comparison of MILP and CP Models for $(F_m|nwt|\sum C_{iM})$ Problem on VRF Instances (10 Machines)

According to the Figure 3.8., the average gap of the CP model stays between 29% and 42% even though the number of jobs is increasing. However, the average gap of the MILP model increases significantly when the number of jobs is increasing.

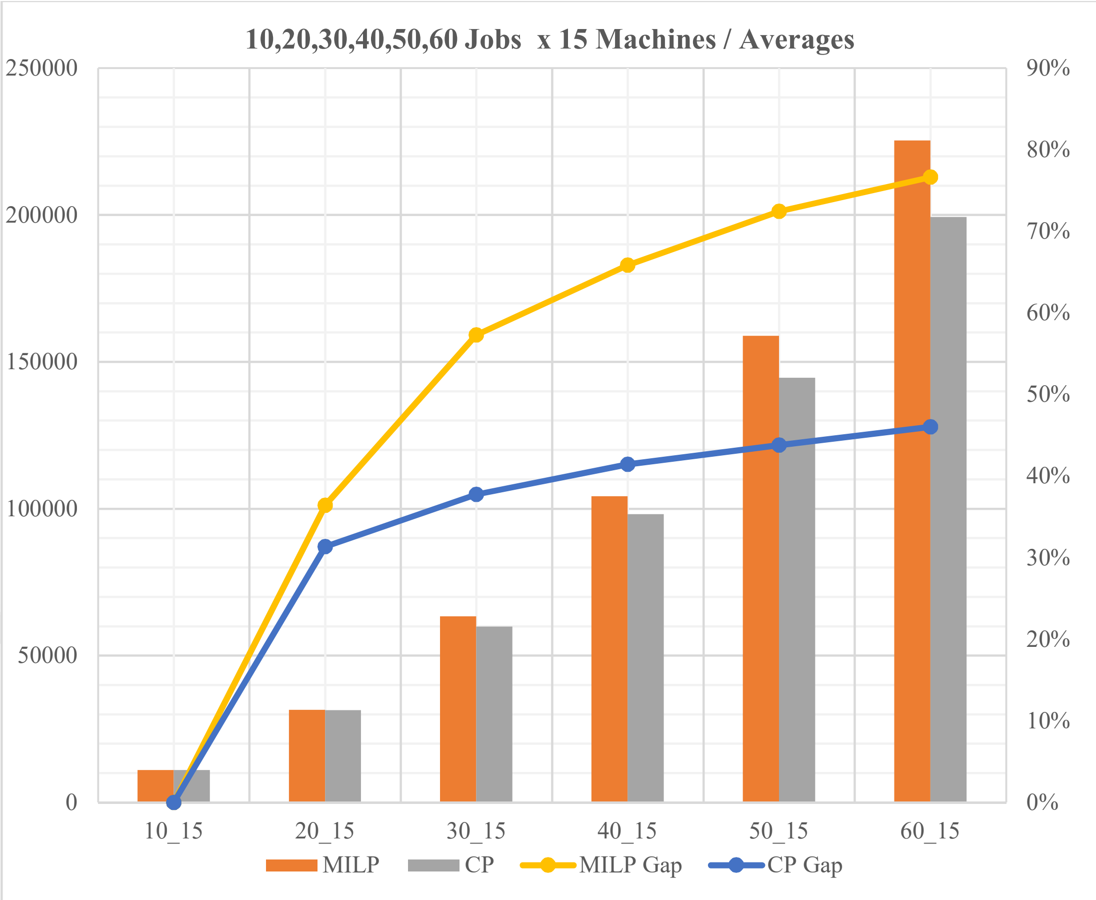


**Figure 3.9.** Comparison of MILP and CP Models for $(F_m|nwt|\sum C_{iM})$ Problem on VRF Instances (15 Machines)

According to the Figure 3.9., the average gap of the CP model stays between 31% and 46% even though the number of jobs is increasing. On the other hand, the average gap of the MILP model rises sharply when the number of jobs is increasing.

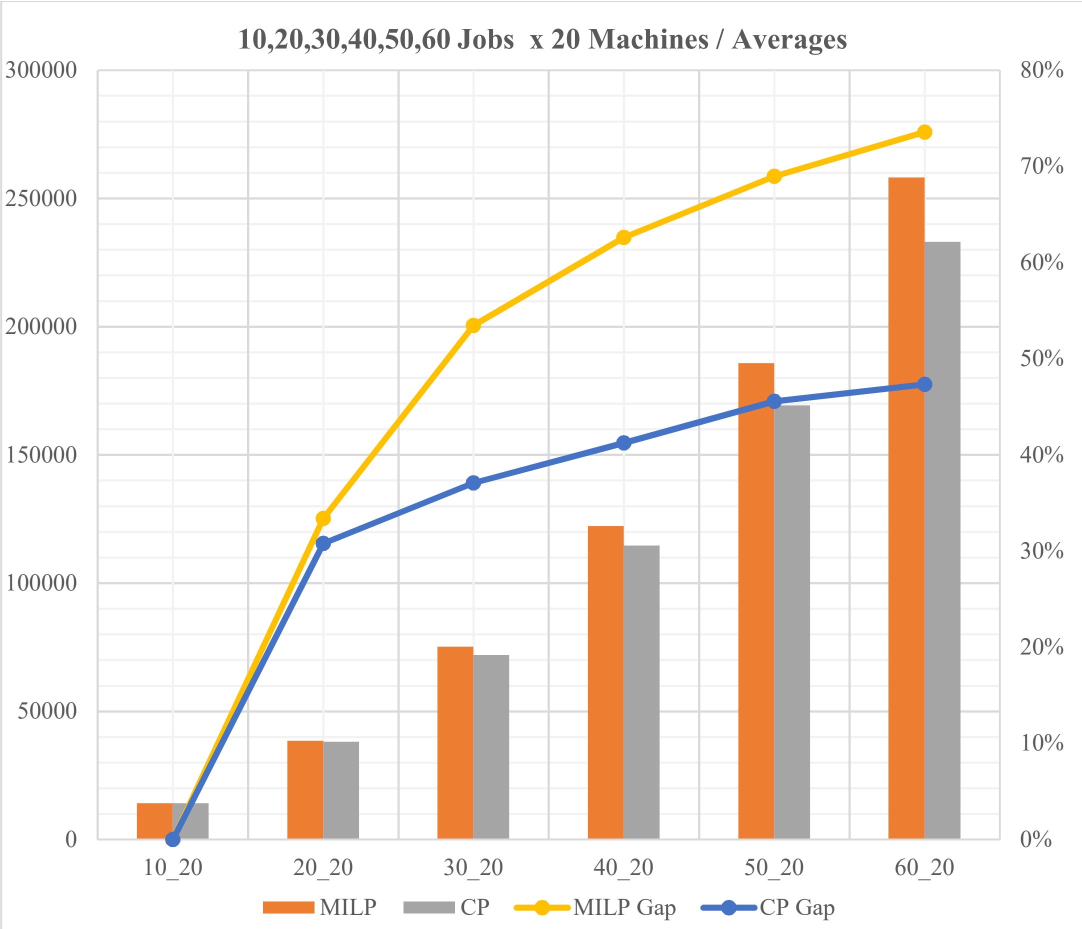


**Figure 3.10.** Comparison of MILP and CP Models for ($F_m|nwt|\sum C_{iM}$) Problem on VRF Instances (20 Machines)

According to the Figure 3.10., the average gap of the CP model stays between 31% and 47% even though the number of jobs is increasing. However, the average gap of the MILP model increases widely when the number of jobs is increasing.

To sum up, if we consider all of the graphs the gap of the CP model is not being affected by the changes in the number of jobs. Its trend follows a similar way when the number of jobs is increasing. In addition, the CP model performs much better than MILP model in terms of the magnitude of the objective function and the gap percentage found within 3600 seconds.

Secondly, the proposed MILP and CP models are executed 3600 seconds on the instances of Taillard (1993) with 11 different combinations of *20x5, 20x10, 20x20, 50x5, 50x10, 50x20, 100x5, 100x10, 100x20, 200x10* and *200x20* set of instances where the first number indicates the number of jobs and the second indicates the number of machines. There are 10 instances of each combination so that 110 instances are studied. All the objective function results are reported in Appendix B (Table B.5.,

Table B.6., Table B.7, Table B.8.). According to these results, like Vallada instances, the CP model outperforms the MILP model. Namely, the gaps of CP is lower than the gaps of MILP, also the objective function values of CP are lower than the objective function of MILP. Another important point here is that while MILP model cannot find any feasible solution for *100x10, 100x20, 200x10* and *200x20* instances, CP model can find a solution for those instances in 3600 seconds.

## 3.3. No-Wait Permutation Flowshop Scheduling Problem with Minimizing Total Tardiness

NWPFSP with the objective of minimizing total flow time $(F_m|nwt|\sum T_i)$ aims to find such a sequence of jobs, which minimizes the total amount of tardiness of all jobs without permitting process queue for the jobs between the machines. The tardiness of jobs $T_i$ can be expressed as $T_i = \max\{(C_{iM} - DD_i),0\}$. That is, a job can be tardy if its completion time exceed its due date.

### 3.3.1. Mixed Integer Linear Programming Model

The mixed integer linear programming model for the single objective no-wait permutation flow shop problem for the minimization of total tardiness $(F_m|nwt|\sum T_i)$ is given below:

| **Model 5.** The MILP Model for the $(F_m|nwt|\sum T_i)$ | | |
|---|---|---|
| **Objective** | | |
| Minimize $\sum_{i\in N} T_i$ | | (5-01) |
| **Constraints** | | |
| $C_{i1} \geq P_{i1}$ | $\forall i \in N$ | (5-02) |
| $C_{ir} - C_{i,r-1} \geq P_{ir}$ | $\forall i \in N, \forall r \in M: r \geq 2$ | (5-03) |
| $C_{ir} - C_{kr} + QD_{ik} \geq P_{ir}$ | $\forall i \in N : k > i, \forall r \in M$ | (5-04) |
| $C_{ir} - C_{kr} + QD_{ik} \leq Q - P_{kr}$ | $\forall i \in N : k > i, \forall r \in M$ | (5-05) |
| $C_{ir} - C_{i,r-1} \leq P_{ir}$ | $\forall i \in N, \forall r \in M: r \geq 2$ | (5-06) |
| $T_i \geq C_{iM} - DD_i$ | $\forall i \in N$ | (5-07) |
| $T_i \geq 0$ | $\forall i \in N$ | (5-08) |
| $C_{ir} \geq 0$ | $\forall i \in N, \forall r \in M$ | (5-09) |
| $D_{ik} \in \{0,1\}$ | $\forall i, k \in N : k > i$ | (5-10) |
| $DD_i \geq 0$ | $\forall i \in N$ | (5-11) |

The objective function (5-01) minimizes the total tardiness. Constraint (5-02) provides that the completion time of the jobs is to be at least its processing time on the first machine. Constraint (5-03) assures that the completion time of each job on machine $r$ can only be greater than or equal to the completion time of the job on machine $r - 1$ plus to the processing time of the job on machine $r$. Then, constraint (5-04) and (5-05) provide that job $k$ either follows the job $i$, or precedes the job $i$, in the sequence. Next, it is provided that the completion time of each job on machine $r$ can only be less than or equal to the completion time of the job on machine $r - 1$ plus to the processing time of the job on machine $r$ by constraint (5-06). Hence, this constraint ensures the no-wait requirement together with the constraint (5-03). In other words, the differences between the completion time of each job on machine $r$ and the completion time of the job on machine $r - 1$must be equal to the processing time of the job on machine $r$. Constraint (7) and (8) together provide that (i) if a job is tardy, then the tardiness of the job at least the completion time of the job on the last machine minus its due; (ii) if a job is early, then its tardiness will be 0. Lastly, the sign restrictions and binary variables are given in (5-09), (5-010) and (5-11) . Note that this model is an extension of the permutation flow shop problem of Manne (1960) by adding the no-wait constraint together with tardiness criterion and by considering the total tardiness in the objective function.

### 3.3.2. Constraint Programming Model

The constraint programming model for the single objective no-wait permutation flow shop problem for minimization of total tardiness ($F_m|nwt|\sum T_i$) is given below:

| **Model 6.** The CP Model for the ($F_m|nwt|\sum T_i$) | | |
|---|---|---|
| **Objective** | | |
| Minimize $\sum_{i\in N} T_i$ where | | (6-01) |
| **Constraints** | | |
| ENDATSTART($Job_{ir}$,$Job_{i,r+1}$) | $\forall i \in N, \forall r \in M: r < M$ | (6-02) |
| NOOVERLAP($Mac_r$) | $\forall r \in M$ | (6-03) |
| SAMESEQUENCE($Mac_1, Mac_r$) | $\forall r \in M : 1 < r$ | (6-04) |
| $T_i = \max((\text{ENDOF } (Job_{iM}) - DD_i),0)$ | $\forall i \in N$ | (6-05) |

The objective function (6-01) minimizes the total tardiness. Constraint (6-02) is the no-wait constraint which provides that the job interval of any given job $i$ on the machine $r$ will be end at the starting time of the job interval of the same job $i$ on the machine $r+1$. Constraint (6-03) provides that there cannot be any overlap on the machines which means that each machine can only process one job at a time. Then, the same sequence for the jobs on each machine is preserved by the constraint (6-04). Lastly, in constraint (6-05) tardiness is expressed as the maximum of end of job intervals on the last machine minus the due dates or 0. Note that a job cannot be tardy if it is finished before its due.

This proposed constraint programming model is an original model which contributes value to the literature of the no-wait flow shop scheduling problems which minimizes the total tardiness.

#### 3.3.3. Computational Results and Comparison of the MILP and CP Models

The instance generation for the tardiness objective is very critic. Although there are some instances for PFSPs with due dates (Vallada et al., 2008; Allahverdi & Aldowaisan, 2004; Ruiz & Stützle, 2008) some considers sequence dependent set up times while generating due dates or some creates different due dates to instances while generating the process times over again. However, since the instances of Taillard (1993) have been used up to this section, the same instances are considered in this section as well. Because the due date generation method of Minella et al. (2008), is employed on the Taillard (1993)'s instances and the resulting due dates are presented in the literature. Hence, 11 different combinations of *20x5, 20x10, 20x20, 50x5, 50x10, 50x20, 100x5, 100x10, 100x20, 200x10* and *200x20* set of instances, where the first number indicates the number of jobs and the second indicates the number of machines, are studied on the $(F_m|nwt|\sum T_i)$ problem. These instances can be found in the http://soa.iti.es website. The due date generation method is the following:

$$DD_i = \sum_{r=1}^{M} P_{ir} * (1 + random(0,1) * 3) \qquad \text{(D-01)}$$

Consecutively, the proposed MILP and CP models are executed 3600 seconds on the instances of Taillard (1993) where 110 instances are studied. All the objective function results are reported in Appendix C (Table C.1., Table C.2., Table C.3., Table C.4.). According to these results, similar with $(F_m|nwt|C_{max})$ and $(F_m|nwt|\sum C_{iM})$ problems, the CP model performs superior than the MILP model. However, the observed trend in the $(F_m|nwt|C_{max})$ and $(F_m|nwt|\sum C_{iM})$ problems could not been encountered in the $(F_m|nwt|\sum T_i)$ problem. Also, some other different results have seen. For example, as seen in Appendix C (Table C.1), MILP model solved the set of *20x20* optimally whereas CP model can solve the same set of instances optimally except the *20x20_05* instance which results 95.57% gap, interestingly. Note that, in the $(F_m|nwt|C_{max})$ and $(F_m|nwt|\sum C_{iM})$ problems the set of *20x20* cannot be solved in 3600 seconds. In addition, as seen in Appendix C (Table C.1), MILP model solved the *20x10* instance set with a gap of 48.92%, however CP model has a 98.23% gap. So, CP has a higher gap than MILP on this set of instance. Any of these situations did not encountered in the $(F_m|nwt|C_{max})$ and the $(F_m|nwt|\sum C_{iM})$ problems. The reason of these is that the nature of the due date generation method in those instances. The other thing is that, although MILP model cannot find any feasible solution for *100x10, 100x20, 200x10* and *200x20* instances, the CP model can find a solution for those instances in 3600 seconds. However, this is encountered in the $(F_m|nwt|C_{max})$ and the $(F_m|nwt|\sum C_{iM})$ problems, too.

To sum up, all the computational results of this chapter demonstrates that CP model is good at finding better solutions than MILP model by also having lower level of solution gap. However, these solutions are still not enough to find optimal solutions for all instances. Hence, some lower bounds and upper bounds are required; constructive heuristics or metaheuristics can be candidates for upper bounds.

# CHAPTER 4

# BI-OBJECTIVE NO-WAIT PERMUTATION FLOWSHOP SCHEDULING PROBLEMS

In this chapter, bi-objective no-wait flowshop scheduling problem (BI-OBJ NWPFSP) with the objective of minimizing the makespan, the total flow time and the total tardiness while considering the total energy consumption as the second objective have been studied in Sections 4.1, 4.2 and 4.3 respectively. Namely, $(F_m|nwt|C_{max}, TEC)$, $(F_m|nwt|\sum C_{iM}, TEC)$ and $(F_m|nwt|\sum T_i, TEC)$ problems are focused. Both mixed-integer linear programming (MILP) and constraint programming (CP) model formulations have been developed for each objective, as reported in this chapter. Moreover, the comparison of the models are represented at the end of each section. Resulting from the bi-objective nature of the problem, a non-dominated set of solutions called as Pareto-optimal set is obtained. Therefore, the dominance relationship features are used when solving the energy efficient NWPFSP (Deb, 2001). The dominance relation concept of Deb (2001) is used while obtaining the results for BI-OBJ NWPFSPs and it is as follows:

- **Dominance relation:** A solution $x^i$ dominates another solution $x^j$ if the two following solutions are satisfied (denoted as $x^i \prec x^j$):
  - $\forall p \in 1, \dots, P;\ f_p(x^i) \leq f_p(x^j)$
  - $\exists p \in 1, \dots, P;\ f_p(x^i) < f_p(x^j)$

  A solution $x^i$ weakly dominates another solution $x^j$ if the two following solutions are satisfied (denoted as $x^i \preccurlyeq x^j$) if:
  - $\forall p \in 1, \dots, P;\ f_p(x^i) \leq f_p(x^j)$

  A solution $x^i$ is indifferent to another solution $x^j$ if the two following solutions are satisfied (denoted as $x^i \sim x^j$) if:
  - $\forall p \in 1, \dots, P;\ f_p(x^i) \nleq f_p(x^j)\ \wedge\ f_p(x^j) \nleq f_p(x^i)$
- **Non-dominated set:** Amongst a set of solutions *X*, the non-dominated set of solutions are the elements of the set $X^*$ non-dominated by any element of the set *X*.
- **Pareto-optimal set:** The non-dominated set of the entire feasible search space *S* is called the Pareto-optimal solutions.

Also, in this study, from the solution methods for multi-objective problems, the augmented ε-constraint method is employed, as it generates only Pareto-optimal solutions. (Mavrotas, 2009).

## 4.1. Bi-Objective No-Wait Permutation Flowshop Scheduling Problem with Minimizing Makespan and Total Energy Consumption

BI-OBJ NWPFSP with the objective of minimizing the makespan and the total energy consumption ($F_m|nwt|C_{max}$, TEC) goals to obtain a sequence of jobs providing no-wait conditions within the production environment while both objectives simultaneously.

### 4.1.1. Mixed Integer Linear Programming Model

The mixed integer linear programming model for the bi-objective no-wait permutation flow shop problem for minimization of makespan and total energy consumption ($F_m|nwt|C_{max}$, TEC) is given below:

**Model 7.** The MILP Model for the ($F_m|nwt|C_{max}$, TEC)

**Objectives**

| | | |
|---|---|---|
| Minimize $C_{max}$, Minimize $TEC$ | | (7-01) |
| **Constraints** | | |
| $C_{i1} \geq \sum_{l \in L} \frac{P_{i1} * y_{i1l}}{s_l}$ | $\forall i \in N$ | (7-02) |
| $C_{ir} - C_{i,r-1} \geq \sum_{l \in L} \frac{P_{ir} * y_{irl}}{s_l}$ | $\forall i \in N, \forall r \in M: r \geq 2$ | (7-03) |
| $C_{ir} - C_{kr} + Q * D_{ik} \geq \sum_{l \in L} \frac{P_{ir} * y_{irl}}{s_l}$ | $\forall i \in N : k > i, \forall r \in M$ | (7-04) |
| $C_{ir} - C_{kr} + Q * D_{ik} \leq Q - \sum_{l \in L} \frac{P_{kr} * y_{krl}}{s_l}$ | $\forall i \in N : k > i, \forall r \in M$ | (7-05) |
| $C_{max} \geq C_{iM}$ | $\forall i \in N$ | (7-06) |
| $C_{ir} - C_{i,r-1} \leq \sum_{l \in L} \frac{P_{ir} * y_{irl}}{s_l}$ | $\forall i \in N, \forall r \in M: r \geq 2$ | (7-07) |
| $\sum_{l \in L} y_{irl} = 1$ | $\forall i \in N, \forall r \in M$ | (7-08) |
| $y_{irl} = y_{i,r+1,l}$ | $\forall i \in N, \forall r \in M: r < M, \forall l \in L$ | (7-09) |
| $\theta_r = C_{max} - \sum_{i \in N} \sum_{l \in L} \frac{P_{ir} * y_{irl}}{s_l}$ | $\forall r \in M$ | (7-10) |
| $TEC = \sum_{i \in N} \sum_{r \in M} \sum_{l \in L} \frac{P_{ir} * \tau_r * \lambda_l}{60 s_l} y_{irl} + \sum_{r \in M} \frac{\varphi_r * \tau_r * \theta_r}{60}$ | | (7-11) |
| $y_{irl} \in \{0,1\}$ | $\forall i \in N, \forall r \in M, \forall l \in L$ | (7-12) |
| $C_{ir} \geq 0$ | $\forall i \in N, \forall r \in M$ | (7-13) |

$D_{ik} \in \{0,1\}$ $\forall i, k \in N : k > i$ (7-14)

The objective function (7-01) minimizes makespan and total energy consumption. Constraint (7-02) ensures that the completion time of each job must be at least its processing time on machine 1. Constraint (7-03) provides that the completion time of each job on machine $r$ is at least the sum of completion time of the job on machine $r-1$ and the processing time of the job. Constraint (7-04) and (7-05) together assure that either job $i$ follows job k or job k follows job $i$ in the sequence, but not both at the same time. Constraint (7-06) calculates the maximum of completion times of all jobs on the last machine which is makespan. Next, constraint (7-07) assures that the completion time of each job on machine $r$ can only be less than or equal to the completion time of the job on machine $r-1$ plus to the processing time of the job on machine $r$. Hence, this constraint provides that the no-wait constraint together with the constraint (7-03). In other words, the differences between the completion time of each job on machine $r$ and the completion time of the job on machine $r-1$ must be equal to the processing time of the job on machine $r$. Constraints (7-08) and (7-09) guarantee that one speed level will be chosen for each job as proposed by Mansouri et al. (2016) and each job will have the same speed level on each machine. Then, constraint (7-10) calculates the idle time on all machines. Finally, the total energy consumption is calculated in kilowatt hour by constraint (7-11) as provided by Mansouri et al. (2016). The sign restriction and the binary variables are provided in constraints (7-12), (7-13) and (7-14). Note that this model is an extension of Manne (1960)'s PFSP model by adding the no-wait restriction and by considering total energy consumption in the objective function as the second objective; correspondingly, with the addition of idle time, total energy consumption calculations as well as the speed level assumptions as constraints.

#### 4.1.2. Constraint Programming Model

The constraint programming model for the bi-objective no-wait permutation flow shop problem for minimization of the makespan and the total energy consumption ($F_m|nwt|C_{max}, TEC$) is given below:

**Model 8.** The CP Model for the ($F_m|nwt|C_{max}$, TEC)

**Objectives**

Minimize $max_{i \in N}$(ENDOF($Job_{iM}$)), Minimize TEC (8-01)

**Constraints**

| | | |
|---|---|---|
| $\text{ENDATSTART}(Job_{ir}, Job_{i,r+1})$ | $\forall i \in N, \forall r \in M: r < M$ | (8-02) |
| $\text{NOOVERLAP}(Mac_r)$ | $\forall r \in M$ | (8-03) |
| $\text{SAMESEQUENCE}(Mac_1, Mac_r)$ | $\forall r \in M : 1 < r$ | (8-04) |
| $\text{ALTERNATIVE}(Job_{ir}, \text{all } (l \text{ in } L)\ JobOpt_{irl})$ | $\forall i \in N, \forall r \in M$ | (8-05) |
| $\text{PRESENCEOF}(JobOpt_{irl})= \text{PRESENCEOF}(JobOpt_{i,r+1,l})$ | $\forall i \in N, \forall r \in M: r < M, \forall l \in L$ | (8-06) |
| $C_{max} = max_{i \in N}(\text{ENDOF}(Job_{iM})$ | | (8-07) |
| $\theta_r = C_{max} - \sum_{i \in N} \sum_{l \in L} \text{PRESENCEOF}(JobOpt_{irl}) * \frac{P_{ir}}{s_l}$ | $\forall r \in M$ | (8-08) |
| $TEC = \sum_{i \in N} \sum_{r \in M} \sum_{l \in L} \text{PRESENCEOF}(JobOpt_{irl}) * \frac{P_{ir} * \tau_r * \lambda_l}{60 s_l} + \sum_{r \in M} \frac{\varphi_r * \tau_r * \theta_r}{60}$ | | (8-09) |

The objective function (8-01) minimizes makespan and total energy consumption. Makespan is the maximum of the end of the job intervals on the last machines. Constraint (8-02) is the no-wait constraint which provides that the job interval of any given job *i* on the machine *r* will be end at the starting time of the job interval of the same job $i$ on the machine $r + 1$. Constraint (8-03) assures that there cannot be any overlap on the machines which means that each machine can only process one job at a time. The same sequence for the jobs on each machine is preserved by constraint (8-04). Constraints (8-05) and (8-06) guarantee that one speed level will be chosen for each job and each job will have the same speed level on each machine. Then, constraint (8-08) calculates the idle time on all machines where the makespan is obtained by constraint (8-07). Lastly, the total energy consumption is calculated in kilowatt hour by constraint (8-08) as provided by Mansouri et al. (2016).

This proposed bi-objective constraint programming model is a novel model which adds value to the literature of the energy-efficient no-wait flow shop scheduling problems which minimizes the makespan.

### 4.1.3. Comparison of MILP and CP Models

Initially, the ($F_m|nwt|C_{max}$, TEC) problem is solved by both MILP and CP models for small sized instances *"5 jobs x 5 machines,5 jobs x 10 machines, 5 jobs x 20 machines"* which are truncated by cropping the first 5 jobs of all *"20 jobs x 5* *machines, 20 jobs x 10 machines, 20 jobs x 20 machines"* instances of Taillard (1993).

To show the conflict between the total energy consumption and the makespan, an example of Pareto optimal set is represented by addressing the 5x5_01 instance in Figure 4.1.

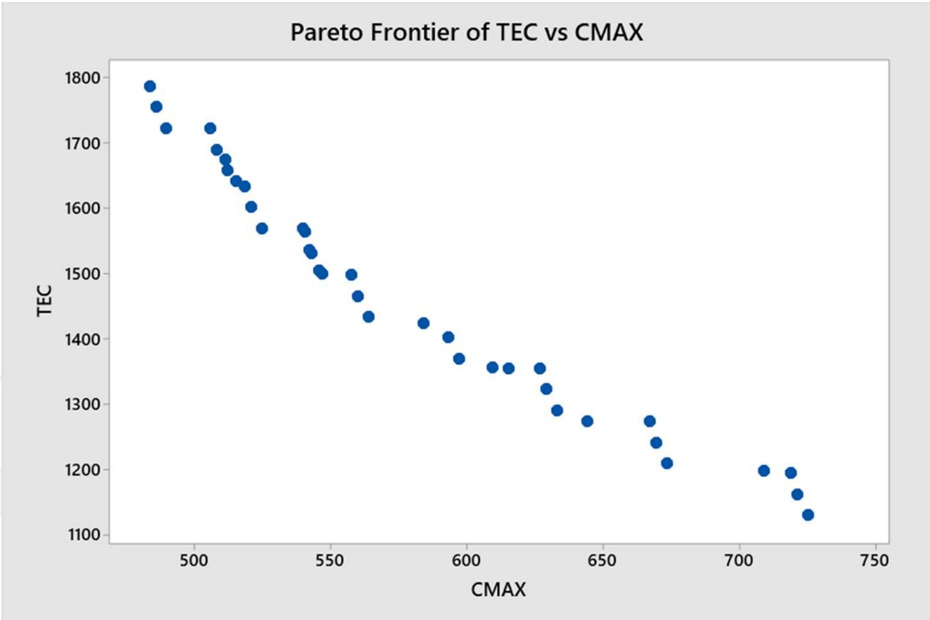


**Figure 4.1.** The Pareto Optimal Set of 5x5_01 for $(F_m|nwt|C_{max}, TEC)$ Problem

As seen in Figure 4.1., two objectives cannot be minimized simultaneously because the total energy consumption increases while the makespan is decreasing and vice versa.

The same pareto optimal set for the makespan and the total energy consumption minimization is obtained for each small sized instance by both MILP and CP model formulations by the augmented ε-constraint method (Mavrotas, 2009), thus the comparison depending on only the computation time of the model formulations can be found in Table 4.1.

**Table 4.1.** MILP and CP Time Comparison for ($F_m|nwt|C_{max}, TEC$) Problem on Small Sized Instances (in seconds)

| Instance Set | MILP | CP | Instance Set | MILP | CP | Instance Set | MILP | CP |
|---|---|---|---|---|---|---|---|---|
| **5x5_01** | 8.48 | 113.43 | **5x10_01** | 4.51 | 125.41 | **5x20_01** | 7.76 | 352.53 |
| **5x5_02** | 6.57 | 73.76 | **5x10_02** | 5.48 | 105.58 | **5x20_02** | 5.53 | 462.84 |
| **5x5_03** | 3.36 | 51.03 | **5x10_03** | 6.14 | 138.57 | **5x20_03** | 4.59 | 518.88 |
| **5x5_04** | 3.79 | 41.05 | **5x10_04** | 4.00 | 109.44 | **5x20_04** | 5.68 | 346.18 |
| **5x5_05** | 3.37 | 44.97 | **5x10_05** | 5.54 | 196.21 | **5x20_05** | 8.32 | 648.80 |
| **5x5_06** | 3.71 | 69.64 | **5x10_06** | 6.25 | 189.99 | **5x20_06** | 4.56 | 418.61 |
| **5x5_07** | 6.18 | 85.56 | **5x10_07** | 4.21 | 184.76 | **5x20_07** | 7.34 | 549.38 |
| **5x5_08** | 4.50 | 52.48 | **5x10_08** | 7.32 | 203.58 | **5x20_08** | 7.29 | 663.32 |
| **5x5_09** | 2.45 | 8.70 | **5x10_09** | 5.23 | 131.13 | **5x20_09** | 5.06 | 555.46 |
| **5x5_10** | 6.82 | 42.16 | **5x10_10** | 7.17 | 178.68 | **5x20_10** | 8.03 | 579.15 |
| **Average** | **4.92** | **58.28** | **Average** | **5.59** | **156.34** | **Average** | **6.42** | **509.52** |

Hence, the superiority of MILP on CP regarding solution time can be seen clearly from Table 4.1. However, since the performance of CPLEX even with 3600 seconds time limit run on single-objective NWPFSPs is not satisfactory, only the proposed heuristics are studied on larger instances.

## 4.2. Bi-Objective No-Wait Permutation Flowshop Scheduling Problem with Minimizing Total Flow Time and Total Energy Consumption

BI-OBJ NWPFSP with the objective of minimizing total flow time and total energy consumption ($F_m|nwt|\sum C_{iM}$, TEC) targets to acquire a permutation of jobs providing no-wait conditions within the production environment while minimizing total flow time and total energy consumption at the same time.

### 4.2.1. Mixed Integer Linear Programming Model

The mixed integer linear programming model for the bi-objective no-wait permutation flow shop problem for minimization of total flow time and total energy consumption ($F_m|nwt|\sum C_{iM}$, TEC) is given below:

**Model 9.** The MILP Model for the ($F_m|nwt|\sum C_{iM}$, TEC)

**Objectives**

| | | |
|---|---|---|
| Minimize$\sum_{i\in N} C_{iM}$, Minimize $TEC$ | | (9-01) |

**Constraints**

| | | |
|---|---|---|
| $C_{i1} \geq \sum_{l\in L} \frac{P_{i1}*y_{i1l}}{s_l}$ | $\forall i \in N$ | (9-02) |
| $C_{ir} - C_{i,r-1} \geq \sum_{l\in L} \frac{P_{ir}*y_{irl}}{s_l}$ | $\forall i \in N, \forall r \in M: r \geq 2$ | (9-03) |
| $C_{ir} - C_{kr} + Q * D_{ik} \geq \sum_{l\in L} \frac{P_{ir}*y_{irl}}{s_l}$ | $\forall i \in N : k > i, \forall r \in M$ | (9-04) |
| $C_{ir} - C_{kr} + Q * D_{ik} \leq Q - \sum_{l\in L} \frac{P_{kr}*y_{krl}}{s_l}$ | $\forall i \in N : k > i, \forall r \in M$ | (9-05) |
| $C_{max} \geq C_{iM}$ | $\forall i \in N$ | (9-06) |
| $C_{ir} - C_{i,r-1} \leq \sum_{l\in L} \frac{P_{ir}*y_{irl}}{s_l}$ | $\forall i \in N, \forall r \in M: r \geq 2$ | (9-07) |
| $\sum_{l\in L} y_{irl} = 1$ | $\forall i \in N, \forall r \in M$ | (9-08) |
| $y_{irl} = y_{i,r+1,l}$ | $\forall i \in N, \forall r \in M: r < M, \forall l \in L$ | (9-09) |
| $\theta_r = C_{max} - \sum_{i\in N}\sum_{l\in L} \frac{P_{ir}*y_{irl}}{s_l}$ | $\forall r \in M$ | (9-10) |
| $TEC = \sum_{i\in N}\sum_{r\in M}\sum_{l\in L} \frac{P_{ir}*\tau_r*\lambda_l}{60 s_l} y_{irl} + \sum_{r\in M} \frac{\varphi_r*\tau_r*\theta_r}{60}$ | | (9-11) |
| $y_{irl} \in \{0,1\}$ | $\forall i \in N, \forall r \in M, \forall l \in L$ | (9-12) |
| $C_{ir} \geq 0$ | $\forall i \in N, \forall r \in M$ | (9-13) |
| $D_{ik} \in \{0,1\}$ | $\forall i, k \in N : k > i$ | (9-14) |

The objective function (9-01) minimizes total flow time and total energy consumption. Constraint (9-02) ensures that the completion time of each job must be at least its processing time on machine 1. Constraint (9-03) provides that the completion time of each job on machine *r* is at least the sum of completion time of the job on machine $r-1$ and the processing time of the job. Constraints (9-04) and (9-05) together assure that either job *i* follows job k or job k follows job *i* in the sequence, but not both at the same time. Constraint (9-06) calculates the maximum of completion times of all jobs on the last machine which is makespan. This constraint is required since makespan is essential for the idle time calculation. Next, constraint (9-07) assures that the completion time of each job on machine *r* can only be less than or equal to the completion time of the job on machine $r-1$ plus to the processing time of the job on machine *r*. Hence, this constraint provides that the no-wait constraint together with the constraint (9-03). In other words, the differences between the completion time of each job on machine $r$ and the completion time of the job on machine $r-1$ must be equal to the processing time of the job on machine $r$. Constraints (9-08) and (9-09) guarantee that one speed level will be chosen for each job and each job will have the same speed level on each machine. Then, constraint (9-10) calculates the idle time on all machines. Finally, the total energy consumption is calculated in kilowatt hour by constraint (9-11) as provided by Mansouri et al. (2016). The sign restriction and the binary variables are provided in constraints (9-12), (9-13) and (9-14). Note that this model is an extension of Manne (1960)'s PFSP model by converting the objective function to total flow time, by adding the no-wait restriction and by considering the total energy consumption in the objective function as the second objective; correspondingly, with the addition of idle time, total energy consumption calculation as well as the speed level assumptions in the model constraints.

### 4.2.2. Constraint Programming Model

The constraint programming model for the bi-objective no-wait permutation flow shop problem for minimization of total flow time and total energy consumption ($F_m|nwt|\sum C_{iM}$, TEC) is given below:

**Model 10.** The CP Model for the ($F_m|nwt|\sum C_{iM}$, TEC)

**Objectives**

| | | |
|---|---|---|
| Minimize $\sum_{i\in N}$ ENDOF($Job_{iM}$), Minimize TEC | | (10-01) |
| **Constraints** | | |
| ENDATSTART($Job_{ir}$, $Job_{i,r+1}$) | $\forall i \in N, \forall r \in M: r < M$ | (10-02) |
| NOOVERLAP($Mac_r$) | $\forall r \in M$ | (10-03) |
| SAMESEQUENCE($Mac_1$, $Mac_r$) | $\forall r \in M : 1 < r$ | (10-04) |
| ALTERNATIVE($Job_{ir}$, all ($l$ in $L$) $JobOpt_{irl}$) | $\forall i \in N, \forall r \in M$ | (10-05) |
| PRESENCEOF($JobOpt_{irl}$)= PRESENCEOF($JobOpt_{i,r+1,l}$) | $\forall i \in N, \forall r \in M$: $r < M, \forall l \in L$ | (10-06) |
| $C_{max} = max_{i\in N}$(ENDOF($JobInt_{iM}$)) | | (10-07) |
| $\theta_r = C_{max} - \sum_{i\in N}\sum_{l\in L}$ PRESENCEOF$(JobOpt_{irl}) * \frac{P_{ir}}{s_l}$ | $\forall r \in M$ | (10-08) |
| $TEC = \sum_{i\in N}\sum_{r\in M}\sum_{l\in L}$ PRESENCEOF$(JobOpt_{irl}) * \frac{P_{ir}*\tau_r*\lambda_l}{60 s_l} + \sum_{r\in M}\frac{\varphi_r*\tau_r*\theta_r}{60}$ | | (10-09) |

The objective function (10-01) minimizes total flow time and total energy consumption. Total flow time is the sum of the end of the job intervals on the last machines. Constraint (10-02) is the no-wait constraint which provides that the job interval of any given job $i$ on the machine $r$ will be end at the starting time of the job interval of the same job $i$ on the machine $r+1$. Constraint (10-03) assures that there cannot be any overlap on the machines which means that each machine can only process one job at a time. The same sequence for the jobs on each machine is preserved by constraint (10-04). Constraints (10-05) and (10-06) guarantee that one speed level will be chosen for each job and each job will have the same speed level on each machine. Then, constraint (10-08) calculates the idle time on all machines where the makespan is obtained by constraint (10-07). Lastly, the total energy consumption is calculated in kilowatt hour by constraint (10-08) as provided by (Mansouri et al., 2016).

This proposed bi-objective constraint programming model is an original model which contributes to the literature of the energy-efficient no-wait flow shop scheduling problems which minimizes total flow time.

### 4.2.3. Comparison of MILP and CP Models

Initially, the ($F_m|nwt|\sum C_{iM}$, TEC) problem is solved by both MILP and CP models for small sized instances *"5 jobs x 5 machines,5 jobs x 10 machines, 5 jobs x 20 machines"* which are truncated by cropping the first 5 jobs of all *"20 jobs x 5 machines, 20 jobs x 10 machines, 20 jobs x 20 machines"* instances of Taillard (1993).

To show the conflict between the total energy consumption and the total flow time, an example of pareto frontier is represented by addressing the instance of 5x5_01 in Figure 4.2. As seen in Figure 4.2., two objectives cannot be minimized simultaneously because the total energy consumption increases while the total flow time is decreasing and vice versa.

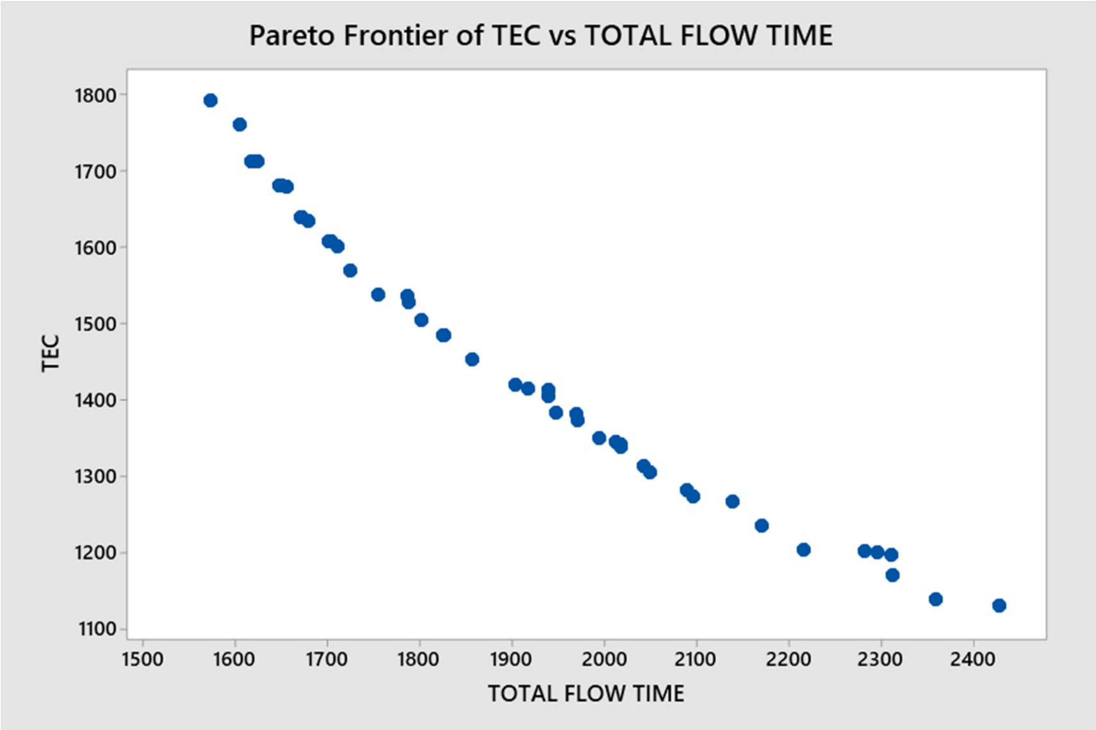


**Figure 4.2.** The Pareto Optimal Set of 5x5_01 for $(F_m|nwt|\sum C_{iM}, TEC)$ Problem

The same pareto optimal set for the total flow time and the total energy consumption is obtained for each small sized instance by both MILP and CP models, thus the comparison depending on only the computation time of the models can be found in Table 4.2.

**Table 4.2.** MILP and CP Time Comparison for $(F_m|nwt|\sum C_{iM}, TEC)$ Problem on Small Sized Instances (in seconds)

| Instance Set | MILP | CP | Instance Set | MILP | CP | Instance Set | MILP | CP |
|---|---|---|---|---|---|---|---|---|
| **5x5_01** | 14.61 | 166.65 | **5x10_01** | 5.40 | 307.26 | **5x20_01** | 17.31 | 1688.73 |
| **5x5_02** | 11.68 | 163.88 | **5x10_02** | 10.26 | 386.06 | **5x20_02** | 11.00 | 1495.02 |
| **5x5_03** | 7.68 | 176.45 | **5x10_03** | 6.04 | 300.84 | **5x20_03** | 14.31 | 2156.56 |
| **5x5_04** | 7.20 | 112.40 | **5x10_04** | 6.20 | 303.77 | **5x20_04** | 9.20 | 1238.21 |
| **5x5_05** | 4.34 | 102.89 | **5x10_05** | 5.54 | 325.91 | **5x20_05** | 16.29 | 1707.89 |
| **5x5_06** | 4.53 | 101.50 | **5x10_06** | 9.98 | 604.54 | **5x20_06** | 11.46 | 1436.78 |
| **5x5_07** | 6.75 | 250.45 | **5x10_07** | 8.26 | 630.87 | **5x20_07** | 10.62 | 1774.45 |
| **5x5_08** | 6.87 | 208.45 | **5x10_08** | 5.59 | 409.74 | **5x20_08** | 13.32 | 1997.10 |
| **5x5_09** | 4.23 | 152.53 | **5x10_09** | 6.96 | 403.35 | **5x20_09** | 11.93 | 1620.52 |
| **5x5_10** | 8.75 | 140.88 | **5x10_10** | 8.79 | 414.53 | **5x20_10** | 11.46 | 1385.07 |
| **Average** | **7.66** | **157.61** | **Average** | **7.30** | **408.69** | **Average** | **12.69** | **1650.03** |

Therefore, the superiority of MILP on CP regarding solution time can be seen clearly from Table 4.2. However, since the performance of CPLEX even with 3600 seconds time limit run on single-objective NWPFSPs is not satisfactory, only the proposed heuristics are studied on larger instances.

## 4.3. Bi-Objective No-Wait Permutation Flowshop Scheduling Problem with Minimizing Total Tardiness and Total Energy Consumption

BI-OBJNWPFSP with the objective of minimizing total tardiness and total energy consumption $(F_m|nwt|\sum T_i)$, TEC) targets to provide a permutation of jobs by ensuring the no-wait requirement of the production environment while minimizing total tardiness and total energy consumption concurrently.

### 4.3.1. Mixed Integer Linear Programming Model

The mixed integer linear programming model for the bi-objective no-wait permutation flow shop problem for minimization of total tardiness and total energy consumption $(F_m|nwt|\sum T_i$, TEC) is given below:

**Model 11.** The MILP Model for the $(F_m|nwt|\sum T_i)$, TEC)

**Objectives**

| | | |
|---|---|---|
| Minimize$\sum_{i\in N} T_i$, Minimize $TEC$ | | (11-01) |

**Constraints**

| | | |
|---|---|---|
| $C_{i1} \geq \sum_{l\in L} \frac{p_{i1} * y_{i1l}}{s_l}$ | $\forall i \in N$ | (11-02) |
| $C_{ir} - C_{i,r-1} \geq \sum_{l\in L} \frac{p_{ir} * y_{irl}}{s_l}$ | $\forall i \in N, \forall r \in M: r \geq 2$ | (11-03) |
| $C_{ir} - C_{kr} + Q * D_{ik} \geq \sum_{l\in L} \frac{p_{ir} * y_{irl}}{s_l}$ | $\forall i \in N : k > i, \forall r \in M$ | (11-04) |
| $C_{ir} - C_{kr} + Q * D_{ik} \leq Q - \sum_{l\in L} \frac{p_{kr} * y_{krl}}{s_l}$ | $\forall i \in N : k > i, \forall r \in M$ | (11-05) |
| $C_{max} \geq C_{iM}$ | $\forall i \in N$ | (11-06) |
| $C_{ir} - C_{i,r-1} \leq \sum_{l\in L} \frac{p_{ir} * y_{irl}}{s_l}$ | $\forall i \in N, \forall r \in M: r \geq 2$ | (11-07) |
| $T_i \geq C_{iM} - DD_i$ | $\forall i \in N$ | (11-08) |
| $T_i \geq 0$ | $\forall i \in N$ | (11-09) |
| $\sum_{l\in L} y_{irl} = 1$ | $\forall i \in N, \forall r \in M$ | (11-10) |
| $y_{irl} = y_{i,r+1,l}$ | $\forall i \in N, \forall r \in M: r < M, \forall l \in L$ | (11-11) |

$$\theta_r = C_{max} - \sum_{i\in N}\sum_{l\in L}\frac{P_{ir} * y_{irl}}{s_l} \qquad \forall r\in M \qquad (11\text{-}12)$$

$$TEC = \sum_{i\in N}\sum_{r\in M}\sum_{l\in L}\frac{P_{ir} * \tau_r * \lambda_l}{60 s_l} y_{irl} + \sum_{r\in M}\frac{\varphi_r * \tau_r * \theta_r}{60} \qquad (11\text{-}13)$$

$$y_{irl} \in \{0,1\} \qquad \forall i\in N, \forall r\in M, \forall l\in L \qquad (11\text{-}14)$$

$$C_{ir} \geq 0 \qquad \forall i\in N, \forall r\in M \qquad (11\text{-}15)$$

$$D_{ik} \in \{0,1\} \qquad \forall i,k\in N : k > i \qquad (11\text{-}16)$$

$$DD_i \geq 0 \qquad \forall i\in N \qquad (11\text{-}17)$$

---

The objective function (11-01) minimizes total tardiness and total energy consumption. Constraint (11-02) ensures that the completion time of each job must be at least its processing time on machine 1. Constraints (11-03) provides that the completion time of each job on machine *r* is at least the sum of completion time of the job on machine *r-1* and the processing time of the job. Constraints (11-04) and (11-05) together assure that either job *i* follows job k or job k follows job *i* in the sequence, but not both at the same time. Constraint (11-06) calculates the maximum of completion times of all jobs on the last machine which is makespan. This constraint is required since makespan is essential for the idle time calculation. Next, constraint (11-07) assures that the completion time of each job on machine *r* can only be less than or equal to the completion time of the job on machine $r-1$ plus to the processing time of the job on machine *r*. Hence, this constraint provides that the no-wait constraint together with the constraint (11-03). In other words, the differences between the completion time of each job on machine *r* and the completion time of the job on machine $r-1$ must be equal to the processing time of the job on machine *r*. Constraints (11-08) and (11-09) together provide that (i) if a job is tardy, then the tardiness of the job at least the completion time of the job on the last machine minus its due; (ii) if a job is early, then its tardiness will be 0. Constraints (11-10) and (11-11) guarantee that one speed level will be chosen for each job and each job will have the same speed level on each machine. Then, constraint (11-12) calculates the idle time on all machines. Finally, the total energy consumption is calculated in kilowatt hour by constraint (11-13) as provided by (Mansouri et al., 2016). The sign restriction and the binary variables are provided in constraints (11-14), (11-15), (11-16) and (11-17). Note that this model is an extension of Manne (1960)'s PFSP model by converting the objective function to total tardiness, by adding the no-wait restriction and by considering the total energy consumption in the objective function as the second objective; correspondingly, with the addition of total tardiness, idle time, total energy consumption calculation as well as the speed level assumptions in the model constraints.

### 4.3.2. Constraint Programming Model

The constraint programming model for the bi-objective no-wait permutation flow shop problem for minimization of total tardiness and total energy consumption ($F_m|nwt|\sum T_i$, TEC) is given below:

**Model 12.** The CP Model for the ($F_m|nwt|\sum T_i$, TEC)

**Objectives**

| | | |
|---|---|---|
| Minimize $\sum_{i\in N} T_i$, Minimize TEC | | (10-01) |
| **Constraints** | | |
| ENDATSTART($Job_{ir}, Job_{i,r+1}$) | $\forall i \in N, \forall r \in M: r < M$ | (10-02) |
| NOOVERLAP($Mac_r$) | $\forall r \in M$ | (10-03) |
| SAMESEQUENCE($Mac_1, Mac_r$) | $\forall r \in M : 1 < r$ | (10-04) |
| $T_i = \max((\text{ENDOF } (JobInt_{iM}) - DD_i), 0)$ | $\forall i \in N$ | (10-05) |
| ALTERNATIVE($Job_{ir}$, all ($l$ in $L$) $JobOpt_{irl}$) | $\forall i \in N, \forall r \in M$ | (10-06) |
| PRESENCEOF($JobOpt_{irl}$)= PRESENCEOF($JobOpt_{i,r+1,l}$) | $\forall i \in N, \forall r \in M$: $r < M, \forall l \in L$ | (10-07) |
| $C_{max} = max_{i\in N}(\text{ENDOF}(JobInt_{iM})$ | | (10-08) |
| $\theta_r = C_{max} - \sum_{i\in N}\sum_{l\in L} \text{PRESENCEOF}(JobOpt_{irl}) * \frac{P_{ir}}{s_l}$ | $\forall r \in M$ | (10-09) |
| $TEC = \sum_{i\in N}\sum_{r\in M}\sum_{l\in L} \text{PRESENCEOF}(JobOpt_{irl}) * \frac{P_{ir}*\tau_r*\lambda_l}{60 s_l} + \sum_{r\in M}\frac{\varphi_r*\tau_r*\theta_r}{60}$ | | (10-10) |

The objective function (12-01) minimizes both the total tardiness and the total energy consumption. A job becomes tardy, if the completion time of the job exceeds its due date. Constraint (12-02) is the no-wait constraint which provides that the job interval of any given job *i* on the machine *r* will be end at the starting time of the job interval of the same job *i* on the machine $r + 1$. Constraint (12-03) assures that there cannot be any overlap on the machines which means that each machine can only process one job at a time. The same sequence for the jobs on each machine is preserved by constraint (12-04). In constraint (12-05), tardiness is expressed as the maximum of end of job intervals on the last machine minus the due dates or 0. Because a job cannot be tardy, if it is finished before its due. Constraints (12-06) and (12-07) guarantee that one speed level will be chosen for each job and each job will have the same speed level on each machine. Then, constraint (12-09) calculates the idle time on all machines where the makespan is obtained by constraint (12-08). Lastly, the total energy consumption is calculated in kilowatt hour by constraint (12-10) as provided by Mansouri et al. (2016).

This proposed bi-objective constraint programming model is a new model which adds value to the literature of the energy-efficient no-wait flow shop scheduling problems which minimizes total tardiness.

#### 4.2.3. Small Sized (Truncated) Instances for ($F_m|nwt|\sum T_i$, TEC) Problem

Total tardiness problem is a very complex problems comparatively to other objectives. Also, there is a limited number of instances in the literature for PFSP which specifies the due dates for each job. Since the instances of Taillard (1993) are employed in this study, the due dates for these instances are taken from Minella et al. (2008). These instances have the due dates $DD_i$ for each job $i \in N$ which $DD_i$ is calculated with the following formulation as given in Section 3.3.3.

$$DD_i = \sum_{r=1}^{M} P_{ir} * (1 + random(0,1) * 3) \qquad \text{(D-01)}$$

In order to have a *5x5, 5x10, and 5x20* sets of instances, *20x5, 20x10 and 20x20* instances are made use of. For the total tardiness objective, each job's individual tardiness is very important. Indeed, the one which creates the total tardiness when adding up together all. Hence, while cropping instances from the jobs view, the aim is the conversation of individual tardiness. Firstly, the schedule for *20x5, 20x10 and 20x20* instances are created as if they are sequenced by PFSP assumptions with the EDD rule. For example, the sequence of 20x5_01 instance is provided in Table 4.3.

**Table 4.3.** EDD Sequence of 20x5_01 Instance Based on PFSP

| Sequence | EDD Sequence | Due Date of Jobs | Cmax of Jobs in EDD | Tardiness of Jobs |
|---|---|---|---|---|
| 1 | 3 | 328 | 126 | 0 |
| 2 | 14 | 482 | 250 | 0 |
| 3 | 19 | 495 | 356 | 0 |
| 4 | 8 | 503 | 397 | 0 |
| 5 | 13 | 525 | 405 | 0 |
| 6 | 17 | 533 | 498 | 0 |
| 7 | 9 | 539 | 567 | 28 |
| 8 | 16 | 568 | 654 | 86 |
| 9 | 5 | 591 | 707 | 116 |
| 10 | 6 | 592 | 769 | 177 |
| 11 | 12 | 602 | 841 | 239 |
| 12 | 20 | 607 | 869 | 262 |
| 13 | 18 | 743 | 941 | 198 |
| 14 | 1 | 767 | 1047 | 280 |
| 15 | 2 | 770 | 1103 | 333 |
| 16 | 7 | 771 | 1156 | 385 |
| 17 | 10 | 805 | 1196 | 391 |
| 18 | 15 | 823 | 1286 | 463 |
| 19 | 11 | 1025 | 1410 | 385 |
| 20 | 4 | 1239 | 1495 | 256 |
| | | **Sum** | **16073** | **3599** |

According to Table 4.3., after having the permutation with EDD rule, the individual tardiness of each job is obtained. After this point, the followed idea is to find the jobs which creates 25% of the total tardiness, since 5 jobs will be truncated from the instance of having 20 jobs.  Herewith, the total tardiness will be truncated by one quarter. If we follow the same example, 16-5-6-12-20 jobs gives 86+116+177+239+262=880 tardiness that is the 24.45% of 3599. Thus, these 5 jobs will be cropped (see Table 4.4.).

**Table 4.4.** EDD Sequence of Cropped 20x5_01 Instance Based on PFSP

| Sequence | EDD Sequence | Due Date of Jobs | Cmax of Jobs in EDD | Tardiness of Jobs |
|---|---|---|---|---|
| 1 | 16 | 568 | 654 | 86 |
| 2 | 5 | 591 | 707 | 116 |
| 3 | 6 | 592 | 769 | 177 |
| 4 | 12 | 602 | 841 | 239 |
| 5 | 20 | 607 | 869 | 262 |
| | | | **Sum** | **880** |

At this point, the completion time of each job $C_{iM}$ and their due dates $DD_i$ are known. Then, only these 5 jobs are sequenced by PFSP assumptions with EDD rule and the results are provided in Table 4.5.

**Table 4.5.** EDD Sequence of 5x5_01 Instance Based on PFSP

| Sequence | EDD Sequence | Due Date of Jobs | Cmax of Jobs in EDD | Tardiness of Jobs |
|---|---|---|---|---|
| 1 | 16 | 568 | 265 | 0 |
| 2 | 5 | 591 | 430 | 0 |
| 3 | 6 | 592 | 503 | 0 |
| 4 | 12 | 602 | 575 | 0 |
| 5 | 20 | 607 | 603 | 0 |

Now, the new completion time of each job $C_{iM}^*$ is known, but the new due dates are required to be created. The main aim is to protect the individual tardiness $T_i$ values. Therefore, the following idea is implemented: $T_i = C_{iM} - DD_i = C_{iM}^* - DD_i^*$ . According to this idea, new due dates can be calculated as: $DD_i^* = C_{iM}^* - C_{iM} + DD_i$. Therefore, the new due dates of the example can be obtained as in Table 4.6.

**Table 4.6.** EDD Sequence of 5x5_01 Instance with New Due Dates

| Sequence | EDD Sequence | Due Date of Jobs | Cmax of Jobs in EDD | Tardiness of Jobs |
|---|---|---|---|---|
| 1 | 16 | **179** | 265 | 86 |
| 2 | 5 | **314** | 430 | 116 |
| 3 | 6 | **326** | 503 | 177 |
| 4 | 12 | **336** | 575 | 239 |
| 5 | 20 | **341** | 603 | 262 |
| | | | **Sum** | **880** |

Hence, the problem generation for small size instances are completed in such a way that individual tardiness values of jobs are preserved. All sets of *5x5, 5x10,* and *5x20* instances are reported in Appendix D (Table D.1.)

### 4.2.4. Comparison of MILP and CP Models

Initially, the ($F_m|nwt|\sum T_i$ ,TEC) problem is solved by both MILP and CP models for small sized instances *"5 jobs x 5 machines,5 jobs x 10 machines, 5 jobs x 20 machines"* which are truncated from the instances of Taillard (1993) by the proposed method described in Section 4.2.3.

To show the conflict between the total energy consumption and the total tardiness, an example of pareto frontier is represented by addressing the instance of 5x5_01 in Figure 4.3. As seen in Figure 4.3., two objectives cannot be minimized simultaneously because the total energy consumption increases while the total tardiness is decreasing and vice versa.

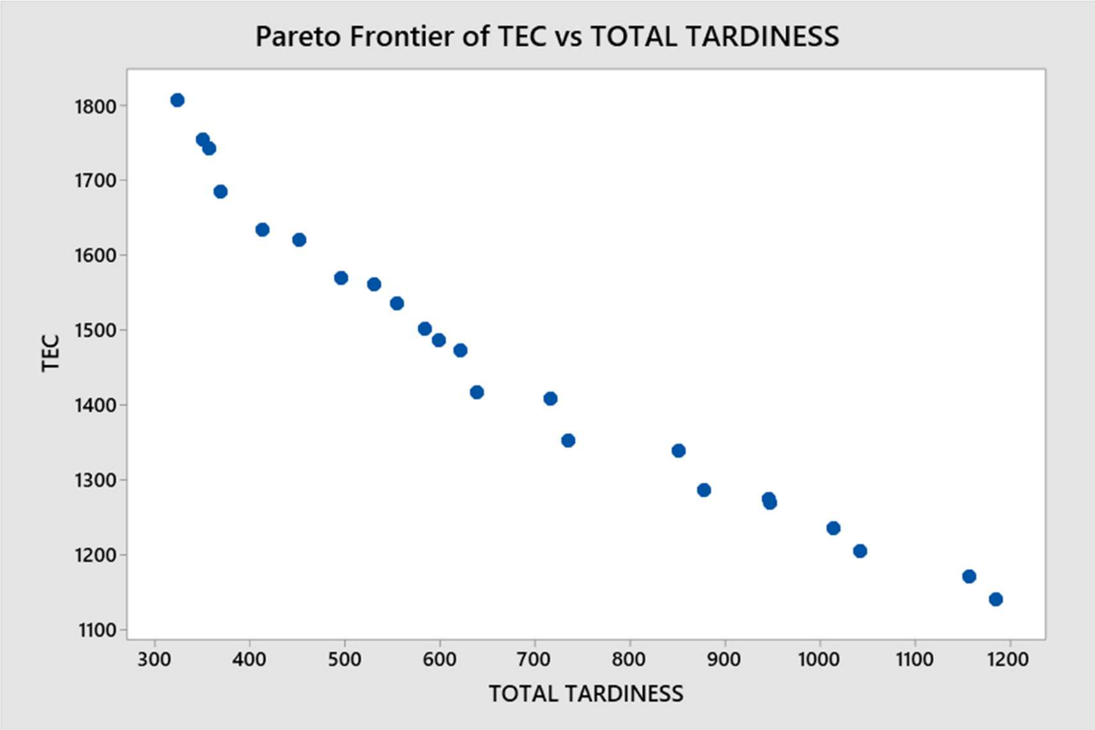


**Figure 4.3.** The Pareto Optimal Set of 5x5_01 for the $(F_m|nwt|\sum T_i, TEC)$ Problem

The same pareto optimal set for makespan and total energy consumption is obtained for each instance by both MILP and CP models, thus the comparison depending on only the computation time of the models can be found in Table 4.7.

**Table 4.7.** MILP and CP Time Comparison for $(F_m|nwt|\sum T_i, TEC)$ Problem on Small Sized Instances (in seconds)

| Instance Set | MILP | CP | Instance Set | MILP | CP | Instance Set | MILP | CP |
|---|---|---|---|---|---|---|---|---|
| **5x5_01** | 6.09 | 101.29 | **5x10_01** | 3.43 | 112.53 | **5x20_01** | 9.93 | 1216.39 |
| **5x5_02** | 10.09 | 136.50 | **5x10_02** | 3.23 | 111.50 | **5x20_02** | 6.29 | 1014.68 |
| **5x5_03** | 5.40 | 137.85 | **5x10_03** | 4.61 | 163.86 | **5x20_03** | 6.82 | 811.27 |
| **5x5_04** | 6.96 | 123.20 | **5x10_04** | 3.81 | 124.15 | **5x20_04** | 2.43 | 251.89 |
| **5x5_05** | 5.37 | 85.21 | **5x10_05** | 5.53 | 197.38 | **5x20_05** | 19.25 | 1228.36 |
| **5x5_06** | 9.34 | 119.84 | **5x10_06** | 8.64 | 373.16 | **5x20_06** | 6.64 | 700.38 |
| **5x5_07** | 10.03 | 206.81 | **5x10_07** | 3.17 | 84.77 | **5x20_07** | 5.71 | 1057.69 |
| **5x5_08** | 6.43 | 144.71 | **5x10_08** | 3.78 | 126.54 | **5x20_08** | 11.15 | 1100.63 |
| **5x5_09** | 13.28 | 305.94 | **5x10_09** | 8.18 | 244.50 | **5x20_09** | 6.56 | 527.76 |
| **5x5_10** | 9.51 | 138.65 | **5x10_10** | 14.04 | 317.97 | **5x20_10** | 5.07 | 368.58 |
| **Average** | **8,25** | **150,00** | **Average** | **5,84** | **185,64** | **Average** | **7,99** | **827,76** |

The superiority of MILP on CP regarding solution time can be seen clearly from Table 4.3. However, since the performance of CPLEX even with 3600 seconds time limit on single-objective NWPFSPs is not satisfactory, only the proposed heuristics those ones explained in the next chapter are studied on larger instances.

# CHAPTER 5
# METAHEURISTICS FOR BI-OBJECTIVE NO-WAIT PERMUTATION FLOWSHOP SCHEDULING PROBLEMS

Regarding the metaheuristic formulation, a significant characteristic of the NWPFSP is employed as proposed by Tasgetiren et al., 2007, Pan et al., 2008, and Wismer, 1972. The difference between the completion time of two consecutive jobs, depends only on the processing times of both jobs, independent from the job's positions in the sequence or the positions of the other jobs in the sequence. Hence, the distance between each combination of two consecutive jobs on the first machine $d[i,j]$ can be calculated as represented in Figure 5.1.

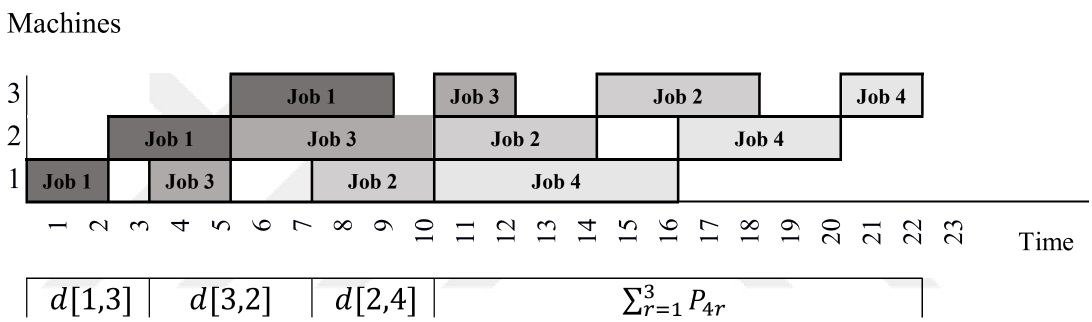


**Figure 5.1.** Gantt Chart for an Example NWPFSP with Distances

$d[i,j] = \{(start\ time\ of\ job\ j\ on\ machine\ 1) - (start\ time\ of\ job\ i\ \ on\ machine\ 1)\}$ if $j$ is processed directly after job $i$.

If this idea is converted to energy efficient scheduling with the speed scaling strategy, then the distance calculation follows the proposed way:

Let a job permutation $\pi = \{\pi_1, \pi_2, \ldots, \pi_N\}$ represent the schedule of jobs to be processed in $m$ machines and $l_\pi = \{l_{\pi_1}, l_{\pi_2}, \ldots, l_{\pi_N}\}$ indicate the speed level of job in the sequence of $\pi$.

Let $d([\pi_{k-1}, \pi_k][l_{\pi_{k-1}}, l_{\pi_k}])$ be the distance between two consecutive jobs in the $k-1^{th}$ and $k^{th}$ positions, $\pi_{k-1}$ and $\pi_k$, whose speed levels are $l_{\pi_{k-1}}$ and $l_{\pi_k}$, respectively. Therefore, these distance values for each pair of job and for each speed level can be written in a $D_{(N)x(N)}L_{(N)x(N)}$ matrix. Thus, there exist 9 $D_{(N)x(N)}L_{(N)x(N)}$ matrices, namely $[\pi_{k-1}, \pi_k][1,1]$, $d[\pi_{k-1}, \pi_k][1,2], \ldots,$ $d[\pi_{k-1}, \pi_k][3,3]$. So, these distances values basically indicate the minimum delay between the start of job $\pi_{k-1}$ and the start of job $\pi_k$ on the first machine if the job $\pi_{k-1}$ directly processed before job $\pi_k$ with speed levels $l_{\pi_{k-1}}$ and $l_{\pi_k}$, respectively, whenever the no-wait restrictions is conserved. The $d([\pi_{k-1},\pi_k][l_{\pi_{k-1}},l_{\pi_k}])$ formulation and the required formulations based on $d([\pi_{k-1},\pi_k][l_{\pi_{k-1}},l_{\pi_k}])$ are given below:

---

**Metaheuristic Formulations.** Metaheuristic Formulations for Bi-Objective Permutation Flowshop Scheduling Problems (For makespan, total flow time and total tardiness)

$$d([\pi_{k-1},\pi_k][l_{\pi_{k-1}},l_{\pi_k}]) = \frac{p_{\pi_{k-1},1}}{s_{l_{\pi_{k-1}}}} + \max\left\{0, \max_{2\le k\le m}\left\{\sum_{h=2}^{k}\frac{p_{\pi_{k-1},h}}{s_{l_{\pi_{k-1}}}} - \sum_{h=1}^{k-1}\frac{p_{\pi_k,h}}{s_{l_{\pi_k}}}\right\}\right\} \quad \text{(H-01)}$$

$\forall k = 2,\dots,N, \forall l \in \{1,2,3\}$

$$C_{\pi_1,M}(\pi_1) = \sum_{r=1}^{M}\frac{P_{\pi_1,r}}{s_{l_{\pi_1}}} \quad \text{(H-02)}$$

$$C_{\pi_k,M}(\pi_k) = \sum_{k=2}^{k} d([\pi_{k-1},\pi_k][l_{k-1},l_k]) + \sum_{r=1}^{M}\frac{P_{\pi_k,r}}{s_{l_{\pi_k}}} \quad \forall k = 2,\dots,N \quad \text{(H-03)}$$

$$C_{max}(\pi) = \sum_{k=2}^{N} d([\pi_{k-1},\pi_k][l_{k-1},l_k]) + \sum_{r=1}^{M}\frac{P_{\pi_N,r}}{s_{l_{\pi_k}}} \quad \text{(H-04)}$$

$$TotalFlowTime(\pi) = \sum_{k=1}^{N} C_{\pi_k,M}(\pi_k) \quad \text{(H-05)}$$

$$TotalTardiness(\pi) = \sum_{k=1}^{N} \max((C_{\pi_k,M}(\pi_k) - DD_{\pi_k}),0) \quad \text{(H-06)}$$

$$\theta_r = C_{max}(\pi) - \sum_{k=1}^{N}\frac{P_{\pi_k,r}}{s_{l_{\pi_k}}} \qquad \forall r = 1,\dots,M \quad \text{(H-07)}$$

$$TEC = \sum_{k=1}^{N}\sum_{r=1}^{M}\frac{P_{\pi_k,r} * \tau_r * \lambda_{l_{\pi_k}}}{60 s_{l_{\pi_k}}} + \sum_{r=1}^{M}\frac{\vartheta_r * \tau_r * \theta_r}{60} \quad \text{(H-08)}$$

---

Equation (H-01) is the minimum delay between the start of job $\pi_{k-1}$ and the start of job $\pi_k$ on the first machine if the job $\pi_k$ directly processed after job $\pi_{k-1}$ with a speed level $l_{\pi_k}$ and $l_{\pi_{k-1}}$, respectively. Equation (H-02) calculates the completion time of the first job in the sequence, that is $\pi_1$. The completion time of the $k^{th}$ job in the sequence, is calculated in Equation (H-03) and it equals to the sum of its processing time and the total delay until the $k^{th}$ job. Thus, makespan is calculated in Equation (H-04) as the sum of the processing time of the last jobs in the sequence and the total delay until the last job. Similarly, for the two other objective functions Equation (H-05) and (H-06) are presented to express the total flow time and total tardiness, respectively. To be consistent with the MILP and CP models given in Chapter 4, idle time and total energy consumption values are calculated in the same way as proposed by Mansouri et al. (2016) in Equation (H-07) and (H-08).

In this thesis, three multi-objective metaheuristic algorithms are proposed: a novel multi-objective discrete artificial bee colony algorithm (MO-DABC), a traditional multi-objective genetic algorithm (MO-GA) and a multi-objective genetic algorithm with a local search (MO-GALS). Firstly, solution representation and initial population are presented in Section 5.1.1, then the MO-DABC, MO-GA and MO-GALS algorithms are provided in the following sections. Next, all these algorithms are studied to minimize three objectives and the results are represented for bi-objective $(F_m|nwt|C_{max}, TEC)$, $(F_m|nwt|\sum C_{iM}, TEC)$ and $(F_m|nwt|\sum T_i, TEC)$ problems in Sections 5.2, 5.3 and 5.4, respectively.

### 5.1. Solution Representation and Initial Population

As it is mentioned in the problem definition, a job-based speed scaling strategy is used for the NWPFSP. Also, similar with the mathematical modelling, the same speed level strategy for all machines is assumed. Therefore, the multi-chromosome structure of Öztop et al. (2018) and Taşgetiren et al. (2018) is used in all algorithms due to the existence of speed level of each job. This structure includes both the permutation for *N* jobs and a speed vector for three speed levels where *L*= {1(fast), 2(normal), 3(slow)}. Hence, an individual $x_i$'s solution can be demonstrated as an example for 5-jobs and 3-speed levels as given in Table 5.1. As shown in Table 5.1, the individual $x^i\left(\pi_k^i, l_{\pi_k}^i\right)$ indicates a solution where the 3rd job is placed in the 1st position ($\pi_1^i = 3$) and its speed level is fast ($l_{\pi_1}^i = 1$); the 2nd job is placed in the 2nd position ($\pi_2^i = 2$) and its speed level slow ($l_{\pi_2}^i = 3$) and so on.

**Table 5.1.** Individual Solution Representation

| | | $\pi_1$ | $\pi_2$ | $\pi_3$ | $\pi_4$ | $\pi_5$ |
|---|---|---|---|---|---|---|
| $x^i\left(\pi_k^i, l_{\pi_k}^i\right)$ | $\pi$ | 3 | 2 | 1 | 4 | 5 |
| | $l$ | 1 | 3 | 2 | 1 | 2 |

Hence, metaheuristic formulations, is mapped one to one as given below:

---

**Metaheuristic Formulations-Example.** An Example for Metaheuristic Formulations for Bi-Objective Permutation Flowshop Scheduling Problems (For the makespan, the total flow time and the total tardiness)

---

$$C_{\pi_1,M}(3) = \sum_{r=1}^{M} \frac{P_{3,r}}{s_1} \quad \text{(H-09)}$$

$$C_{\pi_2,M}(2) = d([3,2][1,3]) + \sum_{r=1}^{M} \frac{P_{2,r}}{s_3} \quad \text{(H-10)}$$

$$C_{\pi_3,M}(1) = d([3,2][1,3]) + d([2,1][3,2]) + \sum_{r=1}^{M} \frac{P_{1,r}}{s_2} \quad \text{(H-11)}$$

$$C_{\pi_4,M}(4) = d([3,2][1,3]) + d([2,1][3,2]) + d([1,4][2,1]) + \sum_{r=1}^{M} \frac{P_{4,r}}{s_1} \quad \text{(H-12)}$$

$$C_{\pi_5,M}(5) = d([3,2][1,3]) + d([2,1][3,2]) + d([1,4][2,1]) + d([4,5][1,2]) + \sum_{r=1}^{M} \frac{P_{5,r}}{s_2} \quad \text{(H-13)}$$

$$C_{max}(\pi) = C_{\pi_5,M}(5) \quad \text{(H-14)}$$

$$TotalFlowTime(\pi) = \sum_{k=1}^{5} C_{\pi_k,M}(\pi_k) = C_{\pi_1,M}(3) + C_{\pi_2,M}(2) + C_{\pi_3,M}(1) + C_{\pi_4,M}(4) + C_{\pi_5,M}(5) \quad \text{(H-15)}$$

$$TotalTardiness(\pi) = \max(C_{\pi_1,M}(3) - DD_{\pi_1}),0) + \max(C_{\pi_2,M}(2) - DD_{\pi_2}),0) + \max(C_{\pi_3,M}(1) - DD_{\pi_3}),0) + \max(C_{\pi_4,M}(4) - DD_{\pi_4}),0) + \max(C_{\pi_5,M}(5) - DD_{\pi_5}),0) \quad \text{(H-16)}$$

---

To generate the initial population with size *NP=10*, FRB5 heuristic of Farahmand Rad et al. (2009), that is an extension of NEH heuristic by Nawaz et al. (1983), is initially used to find an initial solution $\pi^0$, in all of the proposed algorithms (See Figure 5.2.).

$O = DecreasingOrder(\sum_{r=1}^{M} P_{ir})$

$x^0(\pi_1^0) = O_1$

$for\ i = 2\ to\ N\ do$

$\quad x^0(\pi^0) = InsertJobInBestPosition(x^0(\pi^0), O_i)$

$\quad x^0(\pi^0) = ApplyInsertionLocalSearch(x^0(\pi^0))$

$end\ for$

$return\ \pi^0\ with\ n\ jobs$

**Figure 5.2.** FRB5 Constructive Heuristic

FRB5 algorithm initially sort the jobs in a descending order $(O)$ of their total processing times on all machines and the first job in $O$ is chosen to establish a partial solution. Then, the remaining jobs in $O$ are sequentially inserted into the partial solution. Note that, an insertion local search is also applied to the partial solution at each iteration as a local search strategy and it is represented in Figure 5.3.

$for\ all\ jobs\ in\ the\ individual$

$\quad (\pi_j^*) = Remove\ the\ job\ at\ position\ j\ from\ x^i$

$\quad x^*(\pi^*) = InsertInBestPosition\left(x^i(\pi_j^*)\right)$

$\quad if\ \left(f(x^*) \succ f(x^i)\right)\ then\ do$

$\quad\quad x^i = x^*$

$\quad end\ if$

$end\ for$

$return\ \ x^i$

**Figure 5.3.** Insertion Local Search

As an example, let's consider that we have a current solution of $x^0(\pi_j^i) = \{3-2-1-4-5\}$ which is sorted in a descending order of their total processing times and we select the first job as $x^0(\pi_1^0) = \{3\}$. Then, the second job is inserted into all possible positions as follow: $x^0(\pi^0) = \{3-2\}$ and $x^0(\pi^0) = \{2-3\}$. The best solution is obtained, say, $x^0(\pi^0) = \{3-2\}$. Then, the insertion local search is applied in such a way that each job inserted into all positions. For example, $x^0(\pi^0) = \{3-2\}$ and $x^0(\pi^0) = \{2-3\}$ are again obtained since there are 2 jobs until this step. Then, if the best solution is $x^0(\pi^0) = \{2-3\}$, the 3rd job in the sequence is inserted into all positions as follows: $x^0(\pi^0) = \{1-2-3\}$, $x^0(\pi^0) = \{2-1-3\}$ and $x^0(\pi^0) = \{2-3-1\}$. Then, let's say the best solution is $x^0(\pi^0) = \{1-2-3\}$. At this point, insertion local search shows its affect with the insertion of each job into each position and results 6 different permutation as follows: $x^0(\pi^0) = \{2-3-1\}$, $x^0(\pi^0) = \{2-1-3\}$, $x^0(\pi^0) = \{3-2-1\}$, $x^0(\pi^0) = \{3-1-2\}$, $x^0(\pi^0) = \{1-2-3\}$ and $x^0(\pi^0) = \{1-3-2\}$. Next, let's say that the best solution is $x^0(\pi^0) = \{3-1-2\}$, then the 4th job in the sequence is inserted into all position as follows: $x^0(\pi^0) = \{4-3-1-2\}$, $x^0(\pi^0) = \{3-4-1-2\}$, $x^0(\pi^0) = \{3-1-4-2\}$ and $x^0(\pi^0) = \{3-1-2-4\}$ and the resulting best solution is, say, $x^0(\pi^0) = \{3-1-2-4\}$. Next, the above mentioned insertion local search is applied on this solution that is each job is inserted into all positions, then we obtain 24 different permutations. Hence, one best solution is obtained among them, say, $x^0(\pi^0) = \{2-4-3-1\}$. This procedure allows the individual to improve itself and have a chance to all jobs to be inserted into all positions. If we repeat all steps for the 5th job, then, we obtain, say, $x^0(\pi^0) = (= \{5-4-2-3-1\}$ as the initial solution.

Then, the population size is fixed to $NP = 1$ with the initial solution $x^0(\pi^0)$ and 25% of total CPU time is dedicated to MO-DABC, MO-GA and MO-GALS algorithms along with the determination of speed levels as $l_{\pi^0} = 2$, to be able obtain a diversified initial population. Hence, three algorithms are run with normal speed level ($l_{\pi^0} = 2$) for all jobs so that an individual $\pi^{best}$ is obtained. After that, the population size is specified. In order to analyze the effect of population size, NP is specified for different levels. In this thesis, $NP$ is set as $NP = 30$ and $NP = 10$, and the computational results for each population size are provided in Chapter 6. Then, by determining fast, normal and slow speed levels to each job in the $\pi^{best}$, the first, second and third individuals are constructed in this new population, respectively. The rest of the population is constructed by determining the random speed level to each job in the $\pi^{best}$. It is important to mention that when the speed level of a job is altered in the permutation, this leads to a different solution. Then, the archive set $\Omega$ which is empty in the beginning of the procedure is updated by the initial population's non-dominated solutions. In further stages, this archive set $\Omega$ is filled by the new non-dominated solutions while the any dominated member of this set is moved out of the set.

### 5.2. Multi-Objective Discrete Artificial Bee Colony Algorithm (MO-DABC)

Recently, researchers' interest on swarm intelligence, that is basically depends on the self-organized systems' collective behavior, has rapidly increasing. A novel artificial bee colony (ABC) algorithm is revealed in Karaboğa & Baştürk (2008) by the inspiration taken from honey bee swarms' particular bright behavior as a unique swarm intelligence example. This algorithm aims the optimization of multi-variable and multi-modal continuous functions in principle. The competitiveness of the ABC algorithm's performance is illustrated by several comparisons, since the algorithm requires less number of control parameters than the other population-based algorithms (Karaboga & Basturk, 2008). A discrete variant of ABC algorithm has recently been implemented on lot-streaming flow shop problem to minimize weighted earliness and tardiness penalties in Pan et al. (2011). However, there is no any study which employs ABC algorithm for NWPFSP with makespan, total flow time and total tardiness objectives separately while considering the total energy consumption as the second objective. Hence, a novel multi-objective discrete artificial bee colony algorithm (DABC) for NWPFSP is proposed for the $(F_m|nwt|C_{max}$, TEC), $(F_m|nwt|\sum C_{iM}$, TEC) and $(F_m|nwt|\sum T_i$, TEC) problems.

MO-DABC consists of three phases: employed bee phase; onlooker bee phase and scout bee phase.

**Employed Bee Phase:**

Food sources are generated by the employed bees in the neighborhood of their current positions in the basic ABC algorithm. Therefore, a destruction and construction (DC) methodology is used in the employed bee phase of the algorithm as proposed for the iterated greedy algorithm (IGA) in Ruiz & Stützle (2006). The pseudocode of IGA is given in Figure 5.4. According to the DC, the destruction size determines the *dS* number of jobs to be removed with their speed levels from each individual $x^i$ in the population. In this study, the destruction size is taken as 2. The removed jobs and the remaining jobs are stored in $x_D^i$ and $x_P^i$, respectively. Thus, an insertion local search with speed levels is now applicable on the partial solution $x_P^i$ as given in Figure 5.5. That is, the first job is removed from the current partial solution, $x_P^i$ and its speed level is randomly changed and then it is inserted into all positions in $x_P^i$. Then, this insertion procedure is repeated for all the jobs until the last job is inserted into all positions with its randomly changed speed level. At the end, the one which is non-dominated will replace the partial solution. Later on, random speed levels are dedicated as $x_D^i\left(\pi_j^i, l_{\pi_j}^i\right) = rand()\%3$ to each job in $x_D^i$. Each job with their speed levels is inserted into partial solution $x_P^i$ for construction one by one in the order that they are destructed. At this point, two partial solutions in the population are compared using the partial dominance rule. Then, a non-dominated solution from $n$ insertion is obtained. After that, the same local search strategy is applied to the individual obtain by the construction. If the new solution $x^*$ dominates the individual $x^i$ in the population, then $x^i$ is replaced by $x^*$. Eventually, the archive set $\Omega$ is updated by the non-dominated solutions.

---

$for\ all\ individuals\ in\ the\ population$

$\quad x_D^i\left(\pi_j^i, l_{\pi_j}^i\right) = Remove\ the\ dS\ of\ jobs\ at\ position\ j\ and\ its\ speed\ from\ x^i$

$//Destruction$

$\quad x_P^i\left(\pi_j^i, l_{\pi_j}^i\right) = Partial\ solution\ after\ removal$

$\quad x_P^{\blacksquare}\left(\pi_j^{\blacksquare}, l_{\pi_j}^{\blacksquare}\right) = InsertionLocalSearchtoPartialSolution\left(x_P^i\left(\pi_j^i, l_{\pi_j}^i\right)\right)$

$\quad l_{\pi_j}^i = rand()\%3\ //randomly\ change\ the\ speed\ levels\ of\ x_D^i\left(\pi_j^i, l_{\pi_j}^i\right)$

$\quad x^{\blacksquare\blacksquare}(\pi^{\blacksquare\blacksquare}, l^{\blacksquare\blacksquare}) = Insert\ \ x_D^i\left(\pi_j^i, l_{\pi_j}^i\right)\ in\ all\ positions\left(x_P^{\blacksquare}\left(\pi_j^{\blacksquare}, l_{\pi_j}^{\blacksquare}\right)\right)$

$//Construction$

$\quad x^*(\pi^*, l^*) = InsertionLocalSearchtoCompleteSolution\left(x^{\blacksquare\blacksquare}(\pi^{\blacksquare\blacksquare}, l^{\blacksquare\blacksquare})\right)$

$\quad if\ \left(f(x^*) \succ f(x^i)\right)\ then\ do$

$\qquad x^i = x^*$

$\quad end\ if$

$end\ for$

$return\ \ x^i$

---

**Figure 5.4.** Iterated Greedy Algorithm

$for\ all\ jobs\ in\ the\ individual$

$\left(\pi_j^*, l_{\pi_j}^*\right) = Remove\ the\ job\ at\ position\ j\ and\ its\ speed\ from\ x^i$

$l_{\pi_j}^* = rand()\%3$

$x^*(\pi^*, l^*) = InsertInDominatingPosition\left(x^i\left(\pi_j^*, l_{\pi_j}^*\right)\right)$

$if\ \left(f(x^*) \succ f(x^i)\right)\ then\ do$

$x^i = x^*$

$end\ if$

$end\ for$

$return\ x^i$

**Figure 5.5.** Insertion Local Search with Speed Levels

As an example, let's consider that we a current solution of $x^i\left(\pi_j^i, l_{\pi_j}^i\right) = \begin{Bmatrix} 3-2-1-4-5 \\ 1-3-2-1-1 \end{Bmatrix}$ and the $dS = 2$. After removing random 2 jobs, two partial solutions are obtained: $x_D^i\left(\pi_k^i, l_{\pi_k}^i\right) = \begin{Bmatrix} 1-5 \\ 2-1 \end{Bmatrix}$ and $x_P^i\left(\pi_k^i, l_{\pi_k}^i\right) = \begin{Bmatrix} 3-2-4 \\ 1-3-1 \end{Bmatrix}$. Note that, firstly the job 1 is destructed and then job 5 is destructed. Next, an insertion local search is implemented on $x_P^i\left(\pi_k^i, l_{\pi_k}^i\right) = \begin{Bmatrix} 3-2-4 \\ 1-3-1 \end{Bmatrix}$ in such a way that each job is removed with its speed level and inserted into all possible positions. Thus, an non-dominated partial solution for $x_P^i\left(\pi_k^i, l_{\pi_k}^i\right)$ is obtained, say, $x_P^{\blacksquare}\left(\pi_j^{\blacksquare}, l_{\pi_j}^{\blacksquare}\right) = \begin{Bmatrix} 4-3-2 \\ 2-3-2 \end{Bmatrix}$. Then, the random speed levels are assigned to the destructed jobs, say, $x_D^i\left(\pi_k^i, l_{\pi_k}^i\right) = \begin{Bmatrix} 1-5 \\ 1-3 \end{Bmatrix}$ so that the construction can be processed. Then, job 1 is inserted into all possible positions, thus the non-dominated partial solution is obtained. Next, job 5 is inserted into all possible positions and then the non-dominated solution is reached by construction, say, $x^{\blacksquare\blacksquare}(\pi^{\blacksquare\blacksquare}, l^{\blacksquare\blacksquare}) = \begin{Bmatrix} 5-3-1-4-2 \\ 3-3-1-2-2 \end{Bmatrix}$. Finally, the same insertion local search is applied on $x^{\blacksquare\blacksquare}(\pi^{\blacksquare\blacksquare}, l^{\blacksquare\blacksquare})$ and then a non-dominated solution $x^*(\pi^*, l^*) = \begin{Bmatrix} 4-2-1-5-3 \\ 1-3-1-3-1 \end{Bmatrix}$ is selected. Eventually, the archive set $\Omega$ is updated with $x^*(\pi^*, l^*)$.

**Onlooker Bee Phase:**

The block insertion heuristic (BIH) that is proposed by Taşgetiren et al. (2016) is used in the onlooker bee phase of MO-DABC for each individual $x^i$. BIH determines a block *bS* of jobs with their speed levels from each individual $x^i$. In this study, the block size is taken as 2. The removed jobs and the remaining jobs are stored in $x_B^i$ and $x_P^i$, respectively. Then, random speed levels are dedicated as $x_B^i$ by $x_B^i\left(\pi_j^i, l_{\pi_j}^i\right) = rand()\%3$ to each job in $x_B^i$. The same insertion local search (see Figure 5.5.) that is used in the employed bee phase is implemented on the partial solution $x_P^i$. The block $x_{i,B}$ is inserted into the partial solution $x_P^i$ for all positions *(n-bS+1)*. The dominance rule is used when comparing two partial solutions. Next, a non-dominated solution $x^*\left(\pi_j^*, l_{\pi_j}^*\right)$ from $n - bS + 1$ insertion is obtained. If the new solution $x^*$ dominates the individual $x^i$ in the population, then $x_i$ is replaced by $x^*$. Eventually, the archive set *Ω* is updated by the non-dominated solutions. The pseudocode of BIH is given in Figure 5.6.

---

$for\ all\ i's\ in\ the\ population$

$\quad x_B^i\left(\pi_j^i, l_{\pi_j}^i\right) = Remove\ the\ block\ of\ jobs\ with\ size\ bS\ from\ x^i$

$\quad x_P^i\left(\pi_j^i, l_{\pi_j}^i\right) = Partial\ solution\ after\ removal$

$\quad x_P^{■}\left(\pi_j^{■}, l_{\pi_j}^{■}\right) = InsertionLocalSearch\left(x_P^i\left(\pi_j^i, l_{\pi_j}^i\right)\right)$

$\quad l_{\pi_j}^i = rand()\%3$ *//randomly change the speed levels of* $x_B^i\left(\pi_j^i, l_{\pi_j}^i\right)$

$\quad x^*(\pi^*, l^*) = Insert\ \ x_B^i\left(\pi_j^i, l_{\pi_j}^i\right) in\ all\ n - bS + 1\ positions\left(x_P^{■}\left(\pi_j^{■}, l_{\pi_j}^{■}\right)\right)$

$\quad if\ \left(f(x^*) \succ f\left(x^i\right)\right)\ then\ do$

$\qquad x^i = x^*$

$\quad end\ if$

$end\ for$

$return\ \ x^i$

---

**Figure 5.6.** Block Insertion Move

As an example, let consider that we a current solution of $x^i\left(\pi_j^i, l_{\pi_j}^i\right) = \begin{Bmatrix} 3-2-1-4-5 \\ 1-3-2-1-1 \end{Bmatrix}$ and the $bS = 2$. After removing a random block with size 2, two partial solutions are obtained: $x_B^i\left(\pi_k^i, l_{\pi_k}^i\right) = \begin{Bmatrix} 2-1 \\ 3-2 \end{Bmatrix}$ and $x_P^i\left(\pi_k^i, l_{\pi_k}^i\right) = \begin{Bmatrix} 3-4-5 \\ 1-1-1 \end{Bmatrix}$. Then, the random speed levels are assigned to, say, $x_B^i\left(\pi_k^i, l_{\pi_k}^i\right) = \begin{Bmatrix} 2-1 \\ 2-3 \end{Bmatrix}$. Later on, an insertion local search is implemented on $x_P^i\left(\pi_k^i, l_{\pi_k}^i\right) = \begin{Bmatrix} 3-4-5 \\ 1-1-1 \end{Bmatrix}$ in such a way that each job is removed with its sped level and inserted into all possible positions.

Thus, an non-dominated partial solution for $x_P^i\left(\pi_k^i, l_{\pi_k}^i\right)$ is obtained, say, $x_P^{\blacksquare}\left(\pi_j^{\blacksquare}, l_{\pi_j}^{\blacksquare}\right) = \begin{Bmatrix} 4-5-3 \\ 1-2-3 \end{Bmatrix}$. At the end, the block $x_B^i\left(\pi_k^i, l_{\pi_k}^i\right) = \begin{Bmatrix} 2-1 \\ 2-3 \end{Bmatrix}$ is inserted into $n - bS + 1$ positions as follows: $x^{\blacksquare}\left(\pi_j^{\blacksquare}, l_{\pi_j}^{\blacksquare}\right) = \begin{Bmatrix} 2-1-4-5-3 \\ 2-3-1-2-3 \end{Bmatrix}$ , $x^{\blacksquare}\left(\pi_j^{\blacksquare}, l_{\pi_j}^{\blacksquare}\right) = \begin{Bmatrix} 4-2-1-5-3 \\ 1-2-3-2-3 \end{Bmatrix}$, $x^{\blacksquare}\left(\pi_j^{\blacksquare}, l_{\pi_j}^{\blacksquare}\right) = \begin{Bmatrix} 4-5-2-1-3 \\ 1-2-2-3-3 \end{Bmatrix}$ and $x^{\blacksquare}\left(\pi_j^{\blacksquare}, l_{\pi_j}^{\blacksquare}\right) = \begin{Bmatrix} 4-5-3-2-1 \\ 1-2-3-2-3 \end{Bmatrix}$. Eventually, the non-dominated one $x^*\left(\pi_k^*, l_{\pi_k}^*\right)$ is selected among them and the archive set $\Omega$ is updated.

**Scout Bee Phase:**

A food source is generated randomly by a scout bee in the predetermined search space of basic ABC algorithm. However, the effectiveness of search is decreased by this generation, because more information is being derived by the best food source in the population. The reason is that the most promising region is the search region around the best food source. Thus, iterated local search (ILS) as presented in Lourenco et al. (2003) is applied for each individual $x^i$ in the scout bee phase. The pseudocode of ILS is given in Figure 5.7. The aim of the ILS is to escape from local minima. To do so, a perturbation is made on each individual $x^i$, by removing *pS* number of jobs with their speed levels and inserting each job to another position in the individual at random. The removed jobs and the remaining jobs are stored in $x_I^i$ and $x_P^i$, respectively. The perturbation size is used as 2. After perturbation, the same insertion local search (see Figure 5.5.) as in the previous phases are implemented to the solution generated. Similarly, the comparison of new solution $x^*$ and the individual $x^i$ is made by the dominance rule and the achieve set $\Omega$ is updated by the non-dominated solutions.

$for\ all\ i's\ in\ the\ population$

$x_I^i\left(\pi_j^i, l_{\pi_j}^i\right) = Remove\ the\ pS\ \#\ of\ jobs\ at\ position\ j\ and\ its\ speed\ from\ x^i$

$x_P^i\left(\pi_j^i, l_{\pi_j}^i\right) = Partial\ solution\ after\ removal$

$x^{\blacksquare}\left(\pi_j^{\blacksquare}, l_{\pi_j}^{\blacksquare}\right) = Insert\ \ x_I^i\left(\pi_j^i, l_{\pi_j}^i\right) in\ random\ positions\left(x_P^i\left(\pi_j^i, l_{\pi_j}^i\right)\right)$

$x^*(\pi^*, l^*) = InsertionLocalSearch\left(x^{\blacksquare}\left(\pi_j^{\blacksquare}, l_{\pi_j}^{\blacksquare}\right)\right)$

$if\ \left(f(x^*) \succ f\left(x^i\right)\right)\ then\ do$

$x^i = x^*$

$end\ if$

$end\ for$

$return\ \ x^i$

**Figure 5.7.** Iterated Local Search

As an example, let consider that we a current solution of $x^i\left(\pi_k^i, l_{\pi_k}^i\right) = \begin{Bmatrix} 3-2-1-4-5 \\ 1-3-2-1-1 \end{Bmatrix}$ and the $pS = 2$. After removing the random 2 jobs, two partial solutions are obtained: $x_I^i\left(\pi_j^i, l_{\pi_j}^i\right) = \begin{Bmatrix} 2-4 \\ 3-1 \end{Bmatrix}$ and $x_P^i\left(\pi_j^i, l_{\pi_j}^i\right) = \begin{Bmatrix} 3-1-5 \\ 1-2-1 \end{Bmatrix}$. Then, each job in $x_I^i\left(\pi_j^i, l_{\pi_j}^i\right)$ is inserted into randomly a position in $x_P^i\left(\pi_j^i, l_{\pi_j}^i\right)$, say job 2 is inserted into the 1st position and job 4 is inserted into the 3rd position. Hence, $x^{\blacksquare}\left(\pi_j^{\blacksquare}, l_{\pi_j}^{\blacksquare}\right) = \begin{Bmatrix} 2-3-1-4-5 \\ 3-1-2-1-1 \end{Bmatrix}$ is obtained. Then, an insertion local search is implemented on $x^i\left(\pi_j^i, l_{\pi_j}^i\right)$ in such a way that each job is removed with its sped level and inserted into all possible positions. Thus, an non-dominated solution for $x^*\left(\pi_j^*, l_{\pi_j}^*\right)$ is obtained, say, $x^*\left(\pi_j^*, l_{\pi_j}^*\right) = \begin{Bmatrix} 5-1-4-3-2 \\ 2-2-1-2-1 \end{Bmatrix}$. At the end, the archive set *Ω* is updated with the non-dominated solution.

The performance of MO-DABC algorithm is highly remarkable to minimize the makespan. Nevertheless, the need for changing speed levels within the algorithm is essential. Hence, a uniform crossover operator is performed for only the speed levels where keeping the same permutation for each individual $x^i$. To perform this, an individual $x^k$ is selected from the population randomly, for each individual $x^i$. Then, based on the probability of crossover, a new solution is constructed in such a way that either using the speed level of $x^i$ or $x^k$. The construction method of the new solution with a uniform crossover rate is as follows:

$$x^*(\pi^*, l^*) = \begin{cases} l^i_{\pi_j}, & if\ r^i_{\pi_j} \leq CR[i] \\ l^k_{\pi_j}, & otherwise \end{cases} \quad j = 1, \dots, N, \tag{H-17}$$

where $r^i_{\pi_j}$ is generated as a random number that is distributed as $Uniform(0,1)$. Also, the probability of crossover that is $CR[i]$ is derived as $Normal(0.5,0.1)$. If $x^i$ is dominated by $x^*$, then $x^i$ is replaced by $x^*$ in the population and the archive set $\Omega$ is updated. The procedure is followed for all individuals.

Then, a mutation strategy is applied for the speed levels of jobs of each individual $x^i$, after the crossover local search. The strategy is as follows:

$$x^i\left(\pi^i_j, l^i_{\pi_j}\right) = \begin{cases} l^i_{\pi_j} = rand()\%3, & if\ r^i_{\pi_j} \leq MR[i] \\ l^i_{\pi_j}, & otherwise \end{cases} \quad j = 1, \dots, N, \tag{H-18}$$

where $r^i_{\pi_j}$ is generated as a random number that is distributed as $Uniform(0,1)$. Also, the probability of mutation that is $MR[i]$ is derived as $Normal(0.05,0.01)$.

### 5.3. Multi-Objective Genetic Algorithm (MO-GA)

Multi-objective genetic algorithm employs only crossover and mutation strategies. In this algorithm, a two-cut PTL crossover operator as proposed by Pan et al. (2008) is applied to each individual through random selection of another individual from the population by considering the speed levels, as well. If the current individual $x^i$ is dominated by the generated offspring $x^*$, then the offspring $x^*$ substitutes $x^i$ and the archive set $\Omega$ is updated. After that, the mutation for the speed levels as it is explained in Equation (H-18) is applied. That is, an individual is selected randomly for each individual and two-cut PTL crossover operator is used. If the new offspring $x^*$ domintas the individual $x^i$ it is substituted by the $x^i$ and the archive set $\Omega$ is updated. At the end, the population is mutated by Equation (H-18). It is significant to express that in MO-GA, insertion local search (Figure 5.5.) and speed crossover (Equation (H-17)) are not implemented whereas these two strategies are implemented in MO-GALS.

## 5.4. Multi-Objective Genetic Algorithm with a Local Search (MO-GALS)

Multi-objective genetic algorithm with a local search (MO-GALS) is a variant of MO-GA where insertion local search as proposed in Figure 5.3. and crossover local search as provided in Equation (H-17) are additively used on the speed levels. Namely, an individual is selected randomly for each individual and two-cut PTL crossover operator is used. If the new offspring $x^*$domintas the individual $x^i$, it is substituted by the $x_i$ and the archive set $\Omega$ is updated. At the end, the population is mutated by Equation (H-18). Particularly, an insertion local search (Figure 5.5.) and speed crossover (Equation H-17) are implemented in MO-GA.

# CHAPTER 6

# COMPUTATIONAL RESULTS FOR BI-OBJECTIVE NO-WAIT PERMUTATION FLOWSHOP SCHEDULING PROBLEMS

A comprehensive computational analysis is carried out on PFSP benchmark instances of Taillard (1993) to evaluate the performance of the algorithms. Initially, since solving NWPFSP is computationally hard, 30 small size instances with 10 instances of each *5x5, 5x10* and *5x20* set are truncated from *20x5, 20x10* and *20x20* instances. Here, the first number specifies the number of jobs and second number specifies the number of machines in this representation. For $(F_m|nwt|C_{max}, TEC)$ and $(F_m|nwt|\sum C_{iM}, TEC)$ problems, the truncation does not affect the objective function analysis, thus only the first five jobs are cropped from *20x5, 20x10* and *20x20* instances and the *5x5, 5x10,* and *5x20* sets are obtained. However, for $(F_m|nwt|\sum T_i, TEC)$ problem, the truncation is very important since each job's tardiness will affect the total tardiness at the end. Therefore, while truncating the jobs, theirs due dates also needs to be truncated somehow. So, an analysis is provided for the truncation of small sized instances for $(F_m|nwt|\sum T_i, TEC)$ problem and it is already presented in Section 4.2.3.

The speed levels are used as $L$= {1(fast), 2(normal), 3(slow)}. Then, the speed factor and conversion factor for processing speed levels are approved as $s_l = \{1.2, 1.0, 0.8\}$ and $\lambda_l = \{1.5, 1.0, 0.6\}$, respectively. Also, the conversion factor for idle time and the power of machines are 0.05 and 60kW, respectively. All these parameters for the energy efficient scheduling are taken from Mansouri et al. (2016). The mathematical model formulation is coded with the augmented ε-constraint method by using the epsilon value as $10^{-2}$ (Mavrotas, 2009) and all instances are run in IBM ILOG CPLEX 12.6.3 on a Core i7, 2.60 GHz, 8 GB RAM computer in Windows operating system.

After the analysis on small size instances, the larger instances are processed. However, due to the computational difficulty of CPLEX in larger instances, the proposed metaheuristic algorithms are studied on the larger instances which are the first 110 instances of Taillard (1993) for PFSP with 10 instances of each *20x5, 20x10, 20x20, 50x5, 50x10, 50x20, 100x5, 100x10, 100x20, 200x10* and *200x20*. The MO-DABC, MO-GA and MO-GALS algorithms are coded in C++ programming language on Microsoft Visual Studio 2013. All large instances are solved on a Core i5, 3.20 GHz, 8 GB RAM computer. For each instance, 10 replications are carried out. In each replication, the algorithms are run for $25nm$ milliseconds for small instances and $50nm$ milliseconds for larger instances, where $n$ is the number of jobs and $m$ is the number of machines.

To test the performance of MO-DABC, MO-GA and MO-GALS algorithms with CPLEX in small sized instances, three performance measures are used:

***(i)*** **ratio of the Pareto-optimal solutions found:** $R_{p_A} = |A \cap P|/|P|$,

***(ii)*** **inverted generational distance:** $IGD_A = \sum_{v \in P} d(v, A)/|P|$, where the minimum Euclidean distance between two solutions is denoted as $d(v, A)$ (Coello et al. 2002).

*If the IGD value is low, it means that set $A$ is very close to set $P$.

***(iii)*** **distribution spacing:** $DS_A = \left[\frac{1}{|A|}\sum_{i \in A}\left(d_i - \bar{d}\right)^2\right]^{1/2}/\bar{d}$ (Tan et al. 2006).

*If the distribution spacing value is low, it means that the solutions in $M$ are evenly scattered.

Note that the set $A$ refers to the non-dominated solution set of the heuristic algorithms (MO-DABC, MO-GA or MO-GALS). However, to distinguish the algorithms, the sets $X$, $Y$, and $Z$ are defined for the non-dominated solution set of the MO-DABC, MO-GA and MO-GALS algorithms, respectively. Also, $P$ refers to pareto optimal set.

To test the performance of algorithms with each other in larger instances, three performance measures are used:

***(i)*** **cardinality of non-dominated solutions**: $|H|$

***(ii)*** **coverage of two sets:** $C(T, H) = |\{h \in H;\ \exists t \in T: t \succcurlyeq h\}|/|H|$ where $C(T, H)$ equals to 1, if some solutions of $T$ weakly dominate all solutions of $H$ (Zitzler et al. 1999).

***(iii)*** **distribution spacing:** $DS_H = \left[\frac{1}{|H|}\sum_{i \in H}\left(d_i - \bar{d}\right)^2\right]^{1/2}/\bar{d}$ (Tan et al. 2006).

Note that $T$ and $H$ refer to the non-dominated solution set of the heuristic algorithms (MO-DABC, MO-GA or MO-GALS).

### 6.1. Computational Results for Bi-Objective No-Wait Permutation Flowshop Scheduling Problem with Minimizing Makespan and Total Energy Consumption

The three multi-objective algorithms are studied on of small instances for $(F_m|nwt|C_{max}, TEC)$ problem and the averages are represented in Table 6.1. The performance of algorithms on each instance is represented in Appendix D (Table D.1.) Note that, the small size instances for $(F_m|nwt|C_{max}, TEC)$ are cropped only truncating the first 5 jobs of the set of *20x5, 20x10, 20x20* instances.

**Table 6.1.** Comparison of MO-DABC(X), MO-GA(Y) and MO-GALS(Z) with CPLEX on Small Sized Instances for $(F_m|nwt|C_{max}, TEC)$ When Population Size is 30.

| | **MO-DABC** | | | **MO-GA** | | | **MO-GALS** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Instance Set** | $R_{p_X}$ | $IGD_X$ | $DS_X$ | $R_{p_Y}$ | $IGD_Y$ | $DS_Y$ | $R_{p_Z}$ | $IGD_Z$ | $DS_Z$ |
| **5x5** | 1.000 | 0.000 | 0.623 | 1.000 | 0.000 | 0.623 | 1.000 | 0.000 | 0.623 |
| **5x10** | 1.000 | 0.000 | 0.817 | 1.000 | 0.000 | 0.817 | 1.000 | 0.000 | 0.817 |
| **5x20** | 1.000 | 0.000 | 0.835 | 1.000 | 0.000 | 0.835 | 1.000 | 0.000 | 0.835 |
| **Average** | **1.000** | **0.000** | **0.758** | **1.000** | **0.000** | **0.758** | **1.000** | **0.000** | **0.758** |

It is seen that MO-DABC, MO-GA and MO-GALS algorithms detects 100% of the Pareto-optimal set that is created by MILP and CP models with the implementation of augmented ε-constraint method. Also, the inverted generational distance is 0. Moreover, the distribution spacing is low which indicates that the solutions in set of solutions found by MO-DABC, MO-GA and MO-GALS algorithms are uniformly distributed.

Then, the proposed algorithms are run on the large instances and the averages for each set of instances are reported in Tables 6.2 and 6.3.

**Table 6.2.** Comparison of MO-DABC, MO-GA and MO-GALS on Larger Instances in terms of Cardinality and Distribution Spacing for $(F_m|nwt|C_{max}, TEC)$ When Population Size is 30.

| **Instance Set** | $\|X\|$ | $\|Y\|$ | $\|Z\|$ | $DS_X$ | $DS_Y$ | $DS_Z$ |
|---|---|---|---|---|---|---|
| **20x5** | 76.80 | 45.60 | 74.00 | 0.799 | 0.959 | 0.782 |
| **20x10** | 67.90 | 27.00 | 57.90 | 0.856 | 1.294 | 0.925 |
| **20x20** | 53.70 | 13.70 | 46.10 | 0.972 | 1.663 | 0.907 |
| **50x5** | 103.20 | 32.70 | 81.70 | 1.211 | 2.344 | 1.105 |
| **50x10** | 76.00 | 15.50 | 47.30 | 1.336 | 2.414 | 1.097 |
| **50x20** | 49.60 | 6.60 | 24.90 | 1.092 | 1.688 | 1.452 |
| **100x5** | 87.10 | 39.30 | 62.20 | 3.824 | 3.461 | 3.316 |
| **100x10** | 65.30 | 15.80 | 38.20 | 1.473 | 2.844 | 2.181 |
| **100x20** | 45.90 | 6.30 | 15.40 | 1.177 | 1.436 | 2.634 |
| **200x10** | 76.80 | 19.10 | 16.50 | 1.647 | 3.337 | 3.406 |
| **200x20** | 49.50 | 6.60 | 13.00 | 1.185 | 1.410 | 2.672 |
| **Average** | **68.30** | **20.70** | **43.40** | **1.420** | **2.080** | **1.860** |

According to the Table 6.2., MO-DABC finds nearly 3.30 and 1.57 times as many non-dominated solutions than MO-GA and MO-GALS, respectively. Furthermore, lower distribution spacing values indicates uniformly distributed set of solutions so that MO-DABC performs better on terms of this metric.

**Table 6.3.** Comparison of MO-DABC, MO-GA and MO-GALS on Larger Instances in terms of Coverage for $(F_m|nwt|C_{max}, TEC)$ When Population Size is 30.

| Instance Set | $C(X,Y)$ | $C(Y,X)$ | $C(X,Z)$ | $C(Z,X)$ | $C(Y,Z)$ | $C(Z,Y)$ |
|---|---|---|---|---|---|---|
| **20x5** | 0.887 | 0.089 | 0.437 | 0.544 | 0.078 | 0.909 |
| **20x10** | 0.991 | 0.035 | 0.584 | 0.444 | 0.053 | 0.982 |
| **20x20** | 1.000 | 0.025 | 0.573 | 0.420 | 0.040 | 1.000 |
| **50x5** | 0.939 | 0.037 | 0.638 | 0.258 | 0.066 | 0.936 |
| **50x10** | 0.841 | 0.063 | 0.609 | 0.223 | 0.099 | 0.819 |
| **50x20** | 0.957 | 0.058 | 0.645 | 0.172 | 0.118 | 0.953 |
| **100x5** | 0.620 | 0.079 | 0.619 | 0.202 | 0.206 | 0.535 |
| **100x10** | 0.543 | 0.056 | 0.560 | 0.224 | 0.176 | 0.437 |
| **100x20** | 0.755 | 0.013 | 0.712 | 0.170 | 0.055 | 0.685 |
| **200x10** | 0.356 | 0.026 | 0.784 | 0.060 | 0.149 | 0.193 |
| **200x20** | 0.603 | 0.007 | 0.763 | 0.099 | 0.071 | 0.531 |
| **Average** | **0.770** | **0.040** | **0.630** | **0.260** | **0.100** | **0.730** |

One important point is that MO-DABC perform prior with respect to the coverage metric because 77% of the solutions of MOGA are weakly dominated by some solutions of MO-DABC. Also, some solutions of MO-DABC dominates the 63% of the solutions of MOGALS. As a result, MO-DABC performs much better than two other algorithms in terms of both quality and cardinality of non-dominated solutions sets.

Besides, to analyze the effect of population size on the MO-DABC algorithm, population size is taken as 10 and all analyses are repeated, and the results are reported in Tables 6.4. and 6.5. All algorithms find the same pareto optimal set for small size instances with the situation when the population size is 30. Then, the further analysis become meaningful.

**Table 6.4.** Comparison of MO-DABC, MO-GA and MO-GALS on Larger Instances in terms of Cardinality and Distribution Spacing for $(F_m|nwt|C_{max}, TEC)$ When Population Size is 10.

| Instance Set | $\|X\|$ | $\|Y\|$ | $\|Z\|$ | $DS_X$ | $DS_Y$ | $DS_Z$ |
|---|---|---|---|---|---|---|
| **20x5** | 85.10 | 42.00 | 80.70 | 0.787 | 0.987 | 0.845 |
| **20x10** | 67.60 | 24.60 | 54.40 | 0.883 | 1.426 | 0.924 |
| **20x20** | 54.50 | 12.30 | 43.00 | 0.901 | 1.643 | 1.094 |
| **50x5** | 121.70 | 31.90 | 88.60 | 1.121 | 1.795 | 1.094 |
| **50x10** | 81.90 | 14.10 | 52.70 | 1.320 | 2.219 | 1.206 |
| **50x20** | 55.40 | 6.50 | 29.20 | 1.112 | 1.593 | 1.282 |
| **100x5** | 87.00 | 33.50 | 96.10 | 3.935 | 3.133 | 3.504 |
| **100x10** | 61.70 | 14.40 | 59.50 | 1.372 | 2.611 | 2.117 |
| **100x20** | 42.00 | 4.90 | 25.80 | 1.220 | 1.012 | 3.148 |
| **200x10** | 64.30 | 16.80 | 33.00 | 1.838 | 2.858 | 3.546 |
| **200x20** | 42.80 | 8.30 | 18.70 | 1.755 | 1.775 | 2.905 |
| **Average** | **69.50** | **19.00** | **52.90** | **1.477** | **1.914** | **1.970** |

According to the Table 6.4., MO-DABC finds nearly 3.66 and 1.31 times as many non-dominated solutions than MOGA and MOGALS, respectively. Furthermore, lower distribution spacing values indicates uniformly distributed set of solutions therefore MO-DABC’s performance is much better in terms of these metrices.

**Table 6.5.** Comparison of MO-DABC, MO-GA and MO-GALS on Larger Instances in terms of Coverage for $(F_m|nwt|C_{max}, TEC)$ When Population Size is 10.

| Instance Set | $C(X,Y)$ | $C(Y,X)$ | $C(X,Z)$ | $C(Z,X)$ | $C(Y,Z)$ | $C(Z,Y)$ |
|---|---|---|---|---|---|---|
| **20x5** | 0.974 | 0.041 | 0.488 | 0.525 | 0.037 | 0.967 |
| **20x10** | 1.000 | 0.025 | 0.604 | 0.425 | 0.045 | 0.992 |
| **20x20** | 1.000 | 0.025 | 0.602 | 0.338 | 0.043 | 1.000 |
| **50x5** | 0.976 | 0.032 | 0.627 | 0.304 | 0.050 | 0.940 |
| **50x10** | 0.951 | 0.036 | 0.666 | 0.174 | 0.091 | 0.908 |
| **50x20** | 0.986 | 0.053 | 0.641 | 0.198 | 0.103 | 0.945 |
| **100x5** | 0.657 | 0.063 | 0.464 | 0.224 | 0.165 | 0.744 |
| **100x10** | 0.556 | 0.049 | 0.560 | 0.192 | 0.150 | 0.653 |
| **100x20** | 0.807 | 0.073 | 0.403 | 0.233 | 0.139 | 0.835 |
| **200x10** | 0.316 | 0.045 | 0.689 | 0.117 | 0.113 | 0.274 |
| **200x20** | 0.531 | 0.069 | 0.654 | 0.116 | 0.193 | 0.529 |
| **Average** | **0.800** | **0.050** | **0.580** | **0.260** | **0.100** | **0.800** |

Significantly, when population size becomes 10, MO-DABC again performs better with respect to the coverage metric because some solutions of MO-DABC dominates 80% of the solutions of MO-GA. Also, 58% of the solutions of MOGALS are weakly dominated by some solutions of MO-DABC. Therefore, there is a small increase in coverage of MO-GA, but there is a decrease in coverage of MO-GALS.

In conclusion, when the population size is taken as 10, the performance of MO-DABC algorithm gets better over MO-GA, while the coverage behavior over MO-GALS decreases slightly.

### 6.2. Computational Results for Bi-Objective No-Wait Permutation Flowshop Scheduling Problem with Minimizing Total Flow Time and Total Energy Consumption

The three multi-objective algorithms are studied on of small instances for $(F_m|nwt|\sum C_{iM}, TEC)$ problem and the averages are represented in Table 6.6. Note that, the small size instances for $(F_m|nwt|\sum C_{iM}, TEC)$ are cropped only truncating the first 5 jobs of the set of *20x5, 20x10* and *20x20* instances.

**Table 6.6.** Comparison of MO-DABC, MO-GA and MO-GALS with CPLEX on Small Sized Instances for $(F_m|nwt|\sum C_{iM}, TEC)$ When Population Size is 30.

| | **DABC** | | | **MOGA** | | | **MOGALS** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Instance Set** | $R_{p_X}$ | $IGD_X$ | $DS_X$ | $R_{p_Y}$ | $IGD_Y$ | $DS_Y$ | $R_{p_Z}$ | $IGD_Z$ | $DS_Z$ |
| 5x5 | 1.000 | 0.000 | 0.810 | 1.000 | 0.000 | 0.810 | 1.000 | 0.000 | 0.810 |
| 5x10 | 1.000 | 0.000 | 0.777 | 1.000 | 0.000 | 0.777 | 1.000 | 0.000 | 0.777 |
| 5x20 | 0.997 | 0.196 | 0.741 | 1.000 | 0.000 | 0.741 | 1.000 | 0.000 | 0.741 |
| **Average** | **0.999** | **0.065** | **0.776** | **1.000** | **0.000** | **0.776** | **1.000** | **0.000** | **0.776** |

It is seen that the DABC algorithm detects on the average 99.9% of the Pareto-optimal set. The average IGD value is 0.065 indicating that very close approximations to the Pareto-optimal set are found by MO-DABC. There is only one point in one instance that is 5x20_03 in which the algorithm did not found the one point from the Pareto-optimal. Also, the distribution spacing is low which indicates that the solutions in set of solutions found by MO-DABC are uniformly distributed. Hence, MO-DABC performs superior performance since it finds 99.9% of the solution in Pareto-optimal set. Both MOGA and MOGALS algorithms show the same results.

Later, the proposed algorithms are run on the large instances and the averages for each set of instances are reported in Tables 6.7 and 6.8.

**Table 6.7.** Comparison of MO-DABC, MO-GA and MO-GALS on Larger Instances in terms of Cardinality and Distribution Spacing for $(F_m|nwt|\sum C_{iM}, TEC)$ When Population Size is 30.

| Instance Set | $\|X\|$ | $\|Y\|$ | $\|Z\|$ | $DS_X$ | $DS_Y$ | $DS_Z$ |
|---|---|---|---|---|---|---|
| **20x5** | 104.60 | 55.90 | 96.60 | 0.897 | 1.217 | 0.929 |
| **20x10** | 97.80 | 39.20 | 90.40 | 0.826 | 0.887 | 0.922 |
| **20x20** | 102.10 | 29.90 | 80.90 | 0.825 | 0.799 | 0.933 |
| **50x5** | 123.50 | 44.10 | 100.10 | 1.285 | 1.173 | 1.590 |
| **50x10** | 119.30 | 27.00 | 74.90 | 1.226 | 1.132 | 1.619 |
| **50x20** | 101.30 | 15.70 | 45.50 | 1.245 | 1.276 | 1.737 |
| **100x5** | 94.70 | 39.00 | 77.50 | 2.628 | 1.110 | 2.221 |
| **100x10** | 85.40 | 20.10 | 51.90 | 1.702 | 1.206 | 2.409 |
| **100x20** | 74.00 | 12.60 | 30.10 | 1.292 | 1.400 | 1.882 |
| **200x10** | 96.60 | 20.50 | 29.90 | 1.569 | 1.385 | 1.989 |
| **200x20** | 72.50 | 11.90 | 19.50 | 1.482 | 1.209 | 1.577 |
| **Average** | **97.44** | **28.72** | **63.39** | **1.360** | **1.160** | **1.620** |

According to Table 6.7., MO-DABC finds nearly 3.39 and 1.53 times as many non-dominated solutions than MO-GA and MO-GALS, respectively. Furthermore, all algorithms have low distribution spacing values which means that they find the uniformly distributed points.

**Table 6.8.** Comparison of MO-DABC, MO-GA and MO-GALS on Larger Instances in terms of Coverage for $(F_m|nwt|\sum C_{iM}, TEC)$ When Population Size is 30.

| Instance Set | $C(X,Y)$ | $C(Y,X)$ | $C(X,Z)$ | $C(Z,X)$ | $C(Y,Z)$ | $C(Z,Y)$ |
|---|---|---|---|---|---|---|
| **20x5** | 0.978 | 0.033 | 0.430 | 0.537 | 0.028 | 0.970 |
| **20x10** | 0.998 | 0.020 | 0.577 | 0.386 | 0.024 | 0.987 |
| **20x20** | 0.996 | 0.016 | 0.663 | 0.340 | 0.024 | 0.992 |
| **50x5** | 0.958 | 0.052 | 0.788 | 0.140 | 0.097 | 0.914 |
| **50x10** | 0.993 | 0.030 | 0.912 | 0.059 | 0.122 | 0.931 |
| **50x20** | 0.995 | 0.040 | 0.828 | 0.103 | 0.159 | 0.949 |
| **100x5** | 0.753 | 0.043 | 0.735 | 0.113 | 0.221 | 0.667 |
| **100x10** | 0.697 | 0.052 | 0.807 | 0.075 | 0.326 | 0.596 |
| **100x20** | 0.760 | 0.068 | 0.794 | 0.118 | 0.222 | 0.812 |
| **200x10** | 0.652 | 0.029 | 0.899 | 0.040 | 0.219 | 0.385 |
| **200x20** | 0.619 | 0.039 | 0.865 | 0.076 | 0.124 | 0.383 |
| **Average** | **0.850** | **0.040** | **0.750** | **0.180** | **0.140** | **0.780** |

In addition, MO-DABC perform prior in terms of the coverage metric because 85% of the solutions of MOGA are weakly dominated by some solutions of MO-DABC. Also, some solutions of MO-DABC dominates the 75% of the solutions of MOGALS. As a result, MO-DABC performs much better than two other algorithms in terms of both quality and cardinality of non-dominated solutions sets.

Additively, population size is taken as 10 to measure the effect of population size and all analyses are repeated, and the results are reported in Table 6.9. and 6.10. All algorithms find the same pareto optimal set for small sized instances with the situation when the population size is 30. Then, the further analysis become meaningful.

**Table 6.9.** Comparison of MO-DABC, MO-GA and MO-GALS on Larger Instances in terms of Cardinality and Distribution Spacing for $(F_m|nwt|\sum C_{iM}, TEC)$ When Population Size is 10.

| Instance Set | $\|X\|$ | $\|Y\|$ | $\|Z\|$ | $DS_X$ | $DS_Y$ | $DS_Z$ |
|---|---|---|---|---|---|---|
| **20x5** | 152.80 | 56.60 | 90.80 | 1.072 | 1.096 | 1.074 |
| **20x10** | 137.50 | 47.90 | 82.00 | 0.968 | 0.947 | 0.858 |
| **20x20** | 138.70 | 35.10 | 76.70 | 1.013 | 0.722 | 0.986 |
| **50x5** | 220.60 | 55.20 | 113.40 | 1.465 | 1.198 | 1.464 |
| **50x10** | 232.60 | 25.00 | 78.70 | 1.407 | 0.944 | 1.298 |
| **50x20** | 201.50 | 12.60 | 48.80 | 1.274 | 0.894 | 1.551 |
| **100x5** | 140.00 | 36.00 | 111.50 | 1.735 | 1.235 | 2.102 |
| **100x10** | 98.90 | 20.10 | 75.60 | 1.503 | 1.450 | 2.165 |
| **100x20** | 77.90 | 11.60 | 39.90 | 1.699 | 1.453 | 1.571 |
| **200x10** | 77.50 | 20.40 | 54.20 | 1.843 | 1.385 | 3.120 |
| **200x20** | 62.60 | 14.10 | 29.80 | 1.738 | 1.176 | 1.695 |
| **Average** | **140.05** | **30.42** | **72.85** | **1.429** | **1.136** | **1.626** |

**Table 6.10.** Comparison of MO-DABC, MO-GA and MO-GALS on Larger Instances in terms of Coverage for $(F_m|nwt|\sum C_{iM}, TEC)$ When Population Size is 10.

| Instance Set | $C(X,Y)$ | $C(Y,X)$ | $C(X,Z)$ | $C(Z,X)$ | $C(Y,Z)$ | $C(Z,Y)$ |
|---|---|---|---|---|---|---|
| **20x5** | 1.000 | 0.000 | 0.872 | 0.072 | 0.011 | 0.975 |
| **20x10** | 1.000 | 0.000 | 0.920 | 0.046 | 0.000 | 0.998 |
| **20x20** | 1.000 | 0.000 | 0.892 | 0.076 | 0.003 | 0.991 |
| **50x5** | 1.000 | 0.000 | 1.000 | 0.000 | 0.318 | 0.615 |
| **50x10** | 1.000 | 0.000 | 1.000 | 0.000 | 0.089 | 0.916 |
| **50x20** | 1.000 | 0.000 | 1.000 | 0.000 | 0.062 | 0.924 |
| **100x5** | 0.979 | 0.023 | 0.844 | 0.037 | 0.119 | 0.913 |
| **100x10** | 0.814 | 0.044 | 0.837 | 0.057 | 0.189 | 0.763 |
| **100x20** | 0.966 | 0.048 | 0.842 | 0.088 | 0.158 | 0.963 |
| **200x10** | 0.596 | 0.036 | 0.826 | 0.063 | 0.214 | 0.502 |
| **200x20** | 0.730 | 0.063 | 0.847 | 0.084 | 0.192 | 0.620 |
| **Average** | **0.917** | **0.019** | **0.898** | **0.048** | **0.123** | **0.835** |

According to Tables 6.9. and 6.10., MO-DABC finds nearly 4.60 and 1.92 times as many non-dominated solutions than MO-GA and MO-GALS, respectively. One important point is that MO-DABC is much better with respect to the coverage metric because 91.70% of the solutions of MO-GA are weakly dominated by some solutions of DABC. Also, 89.80% of the solutions of MO-GALS are weakly dominated by some solutions of MO-DABC. However, distribution spacing indicates that MO-GA is distributed more uniformly than the solutions of MO-DABC and MO-GALS. As a results, MO-DABC performs much better than two other algorithms in terms of both quality and cardinality of non-dominated solutions sets.

To sum up, when the population size is taken as 10, the performance of MO-DABC algorithm improves over MO-GA and MO-GALS algorithms.

### 6.3. Computational Results for Bi-Objective No-Wait Permutation Flowshop Scheduling Problem with Minimizing Total Tardiness and Total Energy Consumption

The three multi-objective algorithms are studied on of small instances for $(F_m|nwt|\sum T_i, TEC)$ problem and the averages for each instance set are represented in Table 6.11. Note that, the small size instances for $(F_m|nwt|\sum T_i, TEC)$ are obtained by the truncation method that is mentioned in Section 4.2.3. Also, the truncated instances are provided in Appendix D (Table D.1.).

**Table 6.11.** Comparison of MO-DABC, MO-GA and MO-GALS with CPLEX on Small Sized Instances for $(F_m|nwt|\sum T_i, TEC)$ When Population Size is 30.

| | DABC | | | MOGA | | | MOGALS | | |
|---|---|---|---|---|---|---|---|---|---|
| **Instance Set** | $R_{p_X}$ | $IGD_X$ | $DS_X$ | $R_{p_Y}$ | $IGD_Y$ | $DS_Y$ | $R_{p_Z}$ | $IGD_Z$ | $DS_Z$ |
| **5x5** | 1.000 | 0.000 | 0.791 | 1.000 | 0.000 | 0.791 | 1.000 | 0.000 | 0.791 |
| **5x10** | 1.000 | 0.000 | 0.570 | 1.000 | 0.000 | 0.570 | 1.000 | 0.000 | 0.570 |
| **5x20** | 1.000 | 0.000 | 0.565 | 1.000 | 0.000 | 0.565 | 1.000 | 0.000 | 0.565 |
| **Average** | **1.000** | **0.000** | **0.642** | **1.000** | **0.000** | **0.642** | **1.000** | **0.000** | **0.642** |

It is seen that MO-DABC, MO-GA and MO-GALS algorithms detects 100% of the Pareto-optimal set that is created by MILP and CP models with the implementation of augmented ε-constraint method. Also, there is no any inverted generational distance which is 0. Also, the distribution spacing is low which indicates that the solutions in set of solutions found by MO-DABC, MO-GA and MO-GALS algorithms are uniformly distributed.

Next, the all proposed algorithms are runned on the large instances and the averages for each set of instances are reported in Tables 6.12 and 6.13.

**Table 6.12.** Comparison of MO-DABC, MO-GA and MO-GALS on Larger Instances in terms of Cardinality and Distribution Spacing for $(F_m|nwt|\sum T_i, TEC)$ When Population Size is 30.

| Instance Set | $\|X\|$ | $\|Y\|$ | $\|Z\|$ | $DS_X$ | $DS_Y$ | $DS_Z$ |
|---|---|---|---|---|---|---|
| **20x5** | 109.50 | 53.30 | 103.30 | 1.121 | 1.176 | 1.254 |
| **20x10** | 94.90 | 43.00 | 80.40 | 1.239 | 1.394 | 1.343 |
| **20x20** | 73.90 | 27.20 | 59.90 | 1.194 | 1.399 | 1.267 |
| **50x5** | 144.30 | 44.60 | 118.80 | 1.545 | 0.984 | 1.859 |
| **50x10** | 130.50 | 24.80 | 122.90 | 1.573 | 1.124 | 1.905 |
| **50x20** | 120.80 | 12.40 | 104.40 | 1.547 | 1.389 | 1.258 |
| **100x5** | 114.60 | 40.10 | 91.10 | 2.814 | 1.082 | 2.595 |
| **100x10** | 104.00 | 21.60 | 67.00 | 1.768 | 1.141 | 2.719 |
| **100x20** | 94.10 | 12.10 | 66.80 | 1.763 | 1.670 | 1.934 |
| **200x10** | 126.60 | 12.20 | 99.00 | 1.856 | 1.085 | 3.274 |
| **200x20** | 116.30 | 12.80 | 90.40 | 1.766 | 1.223 | 2.288 |
| **Average** | **111.80** | **27.60** | **91.30** | **1.653** | **1.242** | **1.973** |

According to Table 6.12., MO-DABC finds nearly 4.05 and 1.22 times as many non-dominated solutions than MOGA and MOGALS, respectively. Furthermore, lower distribution spacing values indicates uniformly distributed set of solutions so that all algorithms perform good in terms of this metric.

**Table 6.13.** Comparison of MO-DABC, MO-GA and MO-GALS on Larger Instances in terms of Coverage for $(F_m|nwt|\sum T_i, TEC)$ When Population Size is 30.

| Instance Set | $C(X,Y)$ | $C(Y,X)$ | $C(X,Z)$ | $C(Z,X)$ | $C(Y,Z)$ | $C(Z,Y)$ |
|---|---|---|---|---|---|---|
| **20x5** | 0.949 | 0.038 | 0.392 | 0.566 | 0.035 | 0.949 |
| **20x10** | 0.953 | 0.035 | 0.343 | 0.652 | 0.024 | 0.976 |
| **20x20** | 0.988 | 0.008 | 0.432 | 0.573 | 0.000 | 1.000 |
| **50x5** | 0.957 | 0.035 | 0.525 | 0.387 | 0.074 | 0.894 |
| **50x10** | 0.962 | 0.029 | 0.505 | 0.345 | 0.065 | 0.919 |
| **50x20** | 0.959 | 0.021 | 0.467 | 0.437 | 0.041 | 0.943 |
| **100x5** | 0.700 | 0.007 | 0.888 | 0.038 | 0.114 | 0.665 |
| **100x10** | 0.625 | 0.012 | 0.729 | 0.102 | 0.121 | 0.585 |
| **100x20** | 0.786 | 0.024 | 0.773 | 0.103 | 0.122 | 0.551 |
| **200x10** | 0.571 | 0.004 | 0.753 | 0.050 | 0.054 | 0.477 |
| **200x20** | 0.608 | 0.034 | 0.658 | 0.255 | 0.088 | 0.347 |
| **Average** | **0.820** | **0.020** | **0.590** | **0.320** | **0.070** | **0.760** |

It is significant to mention that the performance of MO-DABC is superior regarding the coverage metric since 82% of the solutions of MO-GA are weakly dominated by some solutions of MO-DABC. Also, some solutions of MO-DABC dominates the 59% of the solutions of MO-GALS. In brief, MO-DABC performs much better than two other algorithms in terms of both quality and cardinality of non-dominated solutions sets.

Besides, to the effect of population size on the MO-DABC algorithm, population size is taken as 10 and the analyses are repeated, and the results are reported in Tables 6.14. and 6.15. All algorithms find the same pareto optimal set for small size instances with the situation when the population size is 30. Then, the further analysis become meaningful.

**Table 6.14.** Comparison of MO-DABC, MO-GA and MO-GALS on Larger Instances in terms of Cardinality and Distribution Spacing for $(F_m|nwt|\sum T_i, TEC)$ When Population Size is 10.

| Instance Set | $\|X\|$ | $\|Y\|$ | $\|Z\|$ | $DS_X$ | $DS_Y$ | $DS_Z$ |
|---|---|---|---|---|---|---|
| **20x5** | 126.80 | 54.80 | 109.50 | 1.130 | 0.871 | 1.361 |
| **20x10** | 95.90 | 39.20 | 82.90 | 1.268 | 1.182 | 1.39 |
| **20x20** | 70.40 | 23.30 | 70.00 | 1.236 | 1.598 | 1.500 |
| **50x5** | 188.20 | 38.00 | 147.00 | 1.617 | 1.171 | 2.015 |
| **50x10** | 190.90 | 23.50 | 158.70 | 1.333 | 1.156 | 1.532 |
| **50x20** | 167.30 | 12.90 | 131.10 | 1.247 | 1.216 | 1.868 |
| **100x5** | 122.70 | 37.00 | 139.90 | 2.215 | 1.367 | 2.428 |
| **100x10** | 141.70 | 19.60 | 117.70 | 2.257 | 1.194 | 2.598 |
| **100x20** | 116.20 | 12.70 | 105.70 | 1.532 | 1.357 | 1.619 |
| **200x10** | 111.70 | 20.30 | 74.60 | 1.967 | 1.286 | 2.326 |
| **200x20** | 88.20 | 14.50 | 67.00 | 2.231 | 1.195 | 2.429 |
| **Average** | **129.10** | **26.90** | **109.50** | **1.639** | **1.236** | **1.915** |

According to the Table 6.14., MO-DABC finds nearly 4.79 and 1.17 times as many non-dominated solutions than MO-GA and MO-GALS, respectively. Furthermore, lower distribution spacing values indicates uniformly distributed set of solutions so that all algorithms perform good in terms of this metric.

**Table 6.15.** Comparison of MO-DABC, MO-GA and MO-GALS on Larger Instances in terms of Coverage for $(F_m|nwt|\sum T_i, TEC)$ When Population Size is 10.

| Instance Set | $C(X,Y)$ | $C(Y,X)$ | $C(X,Z)$ | $C(Z,X)$ | $C(Y,Z)$ | $C(Z,Y)$ |
|---|---|---|---|---|---|---|
| **20x5** | 0.976 | 0.016 | 0.307 | 0.633 | 0.014 | 0.992 |
| **20x10** | 0.997 | 0.001 | 0.480 | 0.482 | 0.003 | 0.995 |
| **20x20** | 1.000 | 0.000 | 0.508 | 0.472 | 0.000 | 1.000 |
| **50x5** | 0.976 | 0.031 | 0.490 | 0.384 | 0.063 | 0.914 |
| **50x10** | 1.000 | 0.015 | 0.443 | 0.439 | 0.033 | 0.987 |
| **50x20** | 1.000 | 0.008 | 0.564 | 0.389 | 0.026 | 0.992 |
| **100x5** | 0.855 | 0.044 | 0.430 | 0.325 | 0.088 | 0.845 |
| **100x10** | 0.703 | 0.028 | 0.511 | 0.285 | 0.062 | 0.753 |
| **100x20** | 0.866 | 0.024 | 0.551 | 0.306 | 0.069 | 0.814 |
| **200x10** | 0.543 | 0.008 | 0.792 | 0.086 | 0.104 | 0.464 |
| **200x20** | 0.669 | 0.029 | 0.676 | 0.206 | 0.102 | 0.494 |
| **Average** | **0.871** | **0.019** | **0.523** | **0.364** | **0.051** | **0.841** |

Significantly, when population size becomes 10, MO-DABC again better with respect to the coverage metric because 87% of the solutions of MO-GA are weakly dominated by some solutions of MO-DABC. Also, 52% of the solutions of MO-GALS are weakly dominated by some solutions of MO-DABC. Therefore, there is a small increase in coverage of MO-GA, but there is a decrease in coverage of MO-GALS.

To conclude, when the population size is taken as 10, the performance of MO-DABC algorithm gets better over MO-GA, while the coverage behavior over MO-GALS decreases slightly. Also, from the cardinality perspective, MO-DABC is better when the population size is 10.

### 6.4. A Summary of Computational Results for Bi-Objective No-Wait Permutation Flowshop Scheduling Problems

The findings of this thesis show that (when the population size is used as 10);

- **For the $(F_m|nwt|C_{max})$ problem:**

The MO-DABC algorithm performs 3.66 times better than MOGA and 1.31 times better than MOGALS algorithm in terms of cardinality. Furthermore, lower distribution spacing values indicates uniformly distributed set of solutions therefore MO-DABC's performance is much better in terms of these metrices. More significantly, MO-DABC performs better with respect to the coverage metric because some solutions of MO-DABC dominates 80% of the solutions of MO-GA. Also, 58% of the solutions of MOGALS are weakly dominated by some solutions of MO-DABC.

- **For the $(F_m|nwt|\sum C_{iM})$ problem:**

The MO-DABC algorithm performs 4.60 times better than MOGA and 1.92 times better than MOGALS algorithm in terms of cardinality. More significantly, MO-DABC performs better with respect to the coverage metric because some solutions of MO-DABC dominates 91.70% of the solutions of MO-GA. Also, 89.80% of the solutions of MOGALS are weakly dominated by some solutions of MO-DABC. However, distribution spacing indicates that MO-GA is distributed more uniformly than the solutions of MO-DABC and MO-GALS.

- **For the $(F_m|nwt|\sum T_i)$ problem:**

The MO-DABC algorithm performs 4.79 times better than MOGA and 1.17 times better than MOGALS algorithm in terms of cardinality. More significantly, MO-DABC performs better with respect to the coverage metric because some solutions of MO-DABC dominates 87% of the solutions of MO-GA. Also, 52% of the solutions of MOGALS are weakly dominated by some solutions of MO-DABC. However, distribution spacing indicates that MO-GA is distributed more uniformly than the solutions of MO-DABC and MO-GALS.

To sum up, MO-DABC finds more non-dominated solutions than MO-GA and MO-GALS in all objective functions, based on the cardinality and quality of the solution set. Namely, it is a novel multi-objective metaheuristic algorithm proposed for energy-efficient bi-objective NWPFSPs and it shows its superiority on other heuristics clearly.

# CHAPTER 7
# CONCLUSIONS AND FUTURE RESEARCH

In conclusion, the contribution of this thesis can be divided into three-fold: 1) single-objective NWPFSPs, 2) energy-efficient bi-objective NWPFSPs, and 3) the energy-efficient multi objective metaheuristics. After giving a brief introduction to the NWPFSPs and energy efficient scheduling methods, an extensive literature review is represented. After that, the gaps in the literature of NWPFSPs has been discussed and the resulting work is presented.

Firstly, this thesis aims to investigate a new fundamental mathematical modelling for three important single-objective NWPFSPs: $(F_m|nwt|C_{max})$ and $(F_m|nwt|\sum C_{iM})$ and $(F_m|nwt|\sum T_i)$. To start with, MILP model formulations are proposed for all single objective problems and then CP model formulations are constructed. All the existing instances are executed, and the results are analyzed. Next, some valid inequalities are studied for $(F_m|nwt|C_{max})$ problem. At the end of this chapter, it is revealed that CP model formulation is good alternative way for the NWPFSPs. Also, it is seen that some valid inequalities are quite effective on the MILP model formulation, however, the MILP model formulation cannot be superior than CP model formulation even if it is executed with the valid inequalities for $(F_m|nwt|C_{max})$ problem

Secondly, the bi-objective NWPFSP has been studied with respect to three objectives: $(F_m|nwt|C_{max}, TEC)$ , $(F_m|nwt|\sum C_{iM}, TEC)$ and $(F_m|nwt|\sum T_i, TEC)$ . The energy-efficiency concept is conducted at the operational planning level on machines means that speed scaling strategy is applied on the machines. Namely, machines can process at different speed levels such as slow, normal and fast. First, a MILP model is proposed where the Pareto optimal sets are obtained by augmented epsilon constraint method on Taillard's truncated small instances. Then, CP models are constructed which constitutes the same environment with the MILP model. CP models are also solved by augmented epsilon constraint method. In between, since the truncation is significant in tardiness criterion, an original instance truncation method for $(F_m|nwt|\sum T_i, TEC)$ problem is proposed, and then those truncated instances are used in the $(F_m|nwt|\sum T_i, TEC)$ problem. Hence, this thesis aims to investigate a novel fundamental mathematical modelling for three important bi-objective NWPFSPs.

Finally, due to NP-hardness of the problems, three metaheuristics are proposed: MO-DABC, MO-GA, MO-GALS. The performance of the proposed metaheuristic algorithms is initially measured on the small sized instances to show the ability of heuristics to find non-dominated set of solutions. Three algorithms find 100% of the Pareto-optimal solutions which MILP and CP models found. Then, the larger instances are studied with only the MO-DABC, MO-GA, MO-GALS algorithms, since the problem is NP-Hard to run CPLEX on larger instances. The performance of the algorithms are measured in terms of both quality and cardinality. Hence, based on the comparative computational analyses, the proposed novel MO-DABC is a significantly a beneficial algorithm with respect to all objective function criterions.

For future research, speed scaling strategy can be studied in terms of matrix representation with the adaptation of MILP and CP models and heuristics. In that way the different speed level usage for jobs can be applicable. An important future research direction is to develop lower bounds for the objective functions. This is very critical to improve the efficiency of the proposed algorithms. Also, some other bi-objective metaheuristics might be performed to increase the performance of the algorithms. Furthermore, different performance metrics might be used to measure the quality of the solutions. The last but not the least, different objective functions such as maximum earliness, maximum lateness, number of tardy jobs, etc. can be employed within the framework of energy efficient scheduling.

# REFERENCES


Akhshabi, M., Tavakkoli-Moghaddam, R., & Rahnamay-Roodposhti, F. (2014). A hybrid particle swarm optimization algorithm for a no-wait flow shop scheduling problem with the total flow time. *The International Journal of Advanced Manufacturing Technology*, *70*(5–8), 1181–1188. https://doi.org/10.1007/s00170-013-5351-9

Aldowaisan, T., & Allahverdi, A. (2004). New heuristics for m-machine no-wait flowshop to minimize total completion time. *Omega*, *32*(5), 345–352. https://doi.org/10.1016/J.OMEGA.2004.01.004

Aldowaisan, T., & Allahverdi, A. (2012). Minimizing total tardiness in no-wait flowshops. *Foundations of Computing and Decision Sciences*, *37*(3), 149–162. https://doi.org/10.2478/v10209-011-0009-6

Aldowaisan, T., & Allahverdi, A. (2015). No-Wait Flowshops to Minimize Total Tardiness with Setup Times. *Intelligent Control and Automation*, *6*, 38–44. https://doi.org/10.4236/ica.2015.61005

Allahverdi, A., & Aldowaisan, T. (2004). No-wait flowshops with bicriteria of makespan and maximum lateness. *European Journal of Operational Research*, *152*(1), 132–147. https://doi.org/10.1016/S0377-2217(02)00646-X

Chaudhry, I. A., Elbadawi, I. A., Usman, M., Chughtai, M. T., & Chughtai, M. T. (2018). Minimising Total Flowtime in a No-Wait Flow Shop (NWFS) using Genetic Algorithms. *Ingeniería e Investigación*, *38*(3), 68–79. https://doi.org/10.15446/ing.investig.v38n3.75281

Coello, C. A. C., Lamont, G. B., & Van Veldhuizen, D. A. (2002). *Second Edition Evolutionary Algorithms for Solving Multi-Objective Problems*.

Dai, M.; Tang, D.; Giret Boggino, AS.; Salido Gregorio, MA.; Li, W. (2013). Energy-efficient scheduling for a flexible flow-shop using improved genetic-simulated annealing algorithm. https://doi.org/10.1016/j.rcim.2013.04.001

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons.

Deman, J. M. V., & Baker, K. R. (1974). Minimizing Mean Flowtime in the Flow Shop with No Intermediate Queues. *A I I E Transactions*, *6*(1), 28–34. https://doi.org/10.1080/05695557408974929

Ding, J.-Y., Song, S., Gupta, J. N. D., Zhang, R., Chiong, R., & Wu, C. (2015). An improved iterated greedy algorithm with a Tabu-based reconstruction strategy for the no-wait flowshop scheduling problem. *Applied Soft Computing*, *30*, 604–613. https://doi.org/10.1016/J.ASOC.2015.02.006

Fang, K., Uhan, N., Zhao, F., & Sutherland, J. W. (2011). A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. *Journal of Manufacturing Systems*, *30*(4), 234–240. https://doi.org/10.1016/J.JMSY.2011.08.004

Farahmand Rad, S., Ruiz, R., & Boroojerdian, N. (2009). New High Performing Heuristics for Minimizing Makespan in Permutation Flowshops. *Omega*, 37(2):331–45.

Fink, A., & Voß, S. (2003). Solving the continuous flow-shop scheduling problem by metaheuristics. *European Journal of Operational Research*, *151*(2), 400–414. https://doi.org/10.1016/S0377-2217(02)00834-2

Gahm, C., Denz, F., Dirr, M., & Tuma, A. (2016). Energy-efficient scheduling in manufacturing companies: A review and research framework. *European Journal of Operational Research*, *248*(3), 744–757. https://doi.org/10.1016/j.ejor.2015.07.017

Gao, Kai-zhou, Pan, Q., & Li, J. (2011). Discrete harmony search algorithm for the no-wait flow shop scheduling problem with total flow time criterion. *The International Journal of Advanced Manufacturing Technology*, *56*(5–8), 683–692. https://doi.org/10.1007/s00170-011-3197-6

Gao, K.-Z., Pan, Q.-K., Li, J.-Q., Wang, Y.-T., & Liang, J. (2012). A Hybrid Harmony Search Algorithm for the No-Wait Flow-Shop Schdeuling Problems. *Asia-Pacific Journal of Operational Research*, *29*(02), 1250012. https://doi.org/10.1142/S0217595912500121

Gao, Kaizhou, Pan, Q., Suganthan, P. N., & Li, J. (2013). Effective heuristics for the no-wait flow shop scheduling problem with total flow time minimization. *The International Journal of Advanced Manufacturing Technology*, *66*(9–12), 1563–1572. https://doi.org/10.1007/s00170-012-4440-5

Graham, R. L., Lawler, E. L., Lenstra, J. K., & Kan, A. H. G. R. (1979). Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey (pp. 287–326). https://doi.org/10.1016/S0167-5060(08)70356-X

Hall, N. G., & Sriskandarajah, C. (1996). A Survey of Machine Scheduling Problems with Blocking and No-Wait in Process. *Operations Research*, *44*(3), 510–525. https://doi.org/10.1287/opre.44.3.510

Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, 687–697. https://doi.org/10.1016/j.asoc.2007.05.007

Lin, S.-W., & Ying, K.-C. (2016). Optimization of makespan for no-wait flowshop scheduling problems using efficient matheuristics. *Omega*, *64*, 115–125. https://doi.org/10.1016/J.OMEGA.2015.12.002

Lourenco, H. R., Martin, O., & Stuetzle, T. (2003). *Iterated Local Search*. *International Series in Operations Research & Management Science*. Kluwer Academic Publishers.

Lu, C., Gao, L., Li, X., Pan, Q., & Wang, Q. (2017). Energy-efficient permutation flow shop scheduling problem using a hybrid multi-objective backtracking search algorithm. *Journal of Cleaner Production*, *144*(144), 228–238. https://doi.org/10.1016/j.jclepro.2017.01.011

Manne, A. S. (1960). On the Job-Shop Scheduling Problem. *Operations Research*, *8*(2), 219–223. https://doi.org/10.1287/opre.8.2.219

Mansouri, S. A., Aktas, E., & Besikci, U. (2016). Green scheduling of a two-machine flowshop: Trade-off between makespan and energy consumption. *European Journal of Operational Research*, *248*(3), 772–788. https://doi.org/10.1016/J.EJOR.2015.08.064

Mavrotas, G. (2009). Effective implementation of the ε-constraint method in Multi-Objective Mathematical Programming problems. *Applied Mathematics and Computation*, *213*(2), 455–465. https://doi.org/10.1016/J.AMC.2009.03.037

Minella, G., Ruiz, R., & Ciavotta, M. (2008). A Review and Evaluation of Multiobjective Algorithms for the Flowshop Scheduling Problem. *INFORMS Journal on Computing*, *20*(3), 451–471. https://doi.org/10.1287/ijoc.1070.0258

Mouzon, G., Yildirim, M. B., & Twomey, J. (2007). Operational methods for minimization of energy consumption of manufacturing equipment. *International Journal of Production Research*, *45*(18–19), 4247–4271. https://doi.org/10.1080/00207540701450013

Mouzon, G., & Yildirim, M. B. (2008). A framework to minimise total energy consumption and total tardiness on a single machine. *International Journal of Sustainable Engineering*, *1*(2), 105–116. https://doi.org/10.1080/19397030802257236

Nagano, M. S., & Miyata, H. H. (2016). Review and classification of constructive heuristics mechanisms for no-wait flow shop problem. *The International Journal of Advanced Manufacturing Technology*, *86*(5–8), 2161–2174. https://doi.org/10.1007/s00170-015-8209-5

Nawaz, M., Enscore, E. E., & Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, *11*(1), 91–95. https://doi.org/10.1016/0305-0483(83)90088-9

Öztop, H., Fatih Tasgetiren, M., Türsel Eliiyi, D., & Pan, Q.-K. (2018). Green Permutation Flowshop Scheduling: A Trade- off- Between Energy Consumption and Total Flow Time (pp. 753–759). Springer, Cham. https://doi.org/10.1007/978-3-319-95957-3_79

Pan, Q.-K., Tasgetiren, M. F., & Liang, Y.-C. (2008). A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem. *Computers & Operations Research*, *35*(9), 2807–2839. https://doi.org/10.1016/J.COR.2006.12.030

Pan, Q.-K., Wang, L., & Qian, B. (2009). A novel differential evolution algorithm for bi-criteria no-wait flow shop scheduling problems. *Computers & Operations Research*, *36*(8), 2498–2511. https://doi.org/10.1016/j.cor.2008.10.008

Pan, Q.-K., Tasgetiren, M. F., Suganthan, P. N., & Chua, T. J. (2011). A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. *Information Sciences*, *181*(12), 2455–2468. https://doi.org/10.1016/j.ins.2009.12.025

Röck, H. (1984). The Three-Machine No-Wait Flow Shop is NP-Complete. *Journal of the ACM*, *31*(2), 336–345. https://doi.org/10.1145/62.65

Ruiz, R., & Stützle, T. (2006). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH*, *177*, 2033--2049.

Ruiz, R., & Stützle, T. (2008). An Iterated Greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. *European Journal of Operational Research*, *187*(3), 1143–1159. https://doi.org/10.1016/J.EJOR.2006.07.029

Samarghandi, H. (2015). A particle swarm optimisation for the no-wait flow shop problem with due date constraints. *International Journal of Production Research*, *53*(9), 2853–2870. https://doi.org/10.1080/00207543.2015.1007245

Samarghandi, H., & Behroozi, M. (2017). On the exact solution of the no-wait flow shop problem with due date constraints. *Computers & Operations Research*, *81*, 141–159. https://doi.org/10.1016/J.COR.2016.12.013

Sapkal, S. U., & Laha, D. (2013). A heuristic for no-wait flow shop scheduling. *The International Journal of Advanced Manufacturing Technology*, *68*(5–8), 1327–1338. https://doi.org/10.1007/s00170-013-4924-y

Shafaei, R., Moradinasab, N., & Rabiee, M. (2011). *Efficient meta heuristic algorithms to minimize mean flow time in no-wait two stage flow shops with parallel and identical machines*. *International Journal of Management Science and Engineering Management* (Vol. 6).

Taillard, E. (1993). Benchmarks for basic scheduling problems. *European Journal of Operational Research*, *64*(2), 278–285. https://doi.org/10.1016/0377-

2217(93)90182-M
Tan, K. C., Goh, C. K., Yang, Y. J., & Lee, T. H. (2006). *Evolving Better Population Distribution and Exploration in Evolutionary Multi-Objective Optimization*. *European Journal of Operational Research* (Vol. 171). North-Holland.
Tasgetiren, M. F., Pan, Q.-K., Suganthan, P. N., & Liang, Y.-C. (2007). A Discrete Differential Evolution Algorithm for the No-Wait Flowshop Scheduling Problem with Total Flowtime Criterion. In *2007 IEEE Symposium on Computational Intelligence in Scheduling* (pp. 251–258). IEEE. https://doi.org/10.1109/SCIS.2007.367698
Tasgetiren, M., Pan, Q.-K., Kizilay, D., & Gao, K. (2016). A Variable Block Insertion Heuristic for the Blocking Flowshop Scheduling Problem with Total Flowtime Criterion. *Algorithms*, *9*(4), 71. https://doi.org/10.3390/a9040071
Tasgetiren, M. F., Öztop, H., Eliiyi, U., Eliiyi, D. T., & Pan, Q.-K. (2018). Energy-Efficient Single Machine Total Weighted Tardiness Problem with Sequence-Dependent Setup Times (pp. 746–758). Springer, Cham. https://doi.org/10.1007/978-3-319-95930-6_76
Tavakkoli-Moghaddam, R., Rahimi-Vahed, A. R., & Mirzaei, A. H. (2008). Solving a multi-objective no-wait flow shop scheduling problem with an immune algorithm. *The International Journal of Advanced Manufacturing Technology*, *36*(9–10), 969–981. https://doi.org/10.1007/s00170-006-0906-7
Vallada, E., Ruiz, R., & Minella, G. (2008). Minimising total tardiness in the m-machine flowshop problem: A review and evaluation of heuristics and metaheuristics. *Computers & Operations Research*, *35*(4), 1350–1373. https://doi.org/10.1016/j.cor.2006.08.016
Vallada, E., Ruiz, R., & Framinan, J. M. (2015). New hard benchmark for flowshop scheduling problems minimising makespan. *European Journal of Operational Research*, *240*(3), 666–677. https://doi.org/10.1016/J.EJOR.2014.07.033
Wismer, D. A. (1972). Solution of the Flowshop-Scheduling Problem with No Intermediate Queues. *Operations Research*, *20*(3), 689–697. https://doi.org/10.1287/opre.20.3.689
Ying, K.-C., Lin, S.-W., & Wu, W.-J. (2016). Self-adaptive ruin-and-recreate algorithm for minimizing total flow time in no-wait flowshops. *Computers & Industrial Engineering*, *101*, 167–176. https://doi.org/10.1016/J.CIE.2016.08.014
Ying, K.-C., & Lin, S.-W. (2018). Minimizing makespan for no-wait flowshop scheduling problems with setup times. *Computers & Industrial Engineering*, *121*, 73–81. https://doi.org/10.1016/J.CIE.2018.05.030
Zitzler, E., Technische, E., & Zürich, H. (1999). *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*.

# APPENDIX A – Computational Results for $(F_m|nwt|C_{max})$

## Table A.1. MILP and CP Comparison Table for 5 machines (VRF Instances)

| Instance | Opt. | MILP | Time (Seconds) | Gap % | CP | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|---|
| 10_5_01 | 760 | 760 | 0.719 | 0.00% | 760 | 10.628 | 0.00% |
| 10_5_02 | 759 | 759 | 0.531 | 0.00% | 759 | 4.431 | 0.00% |
| 10_5_03 | 823 | 823 | 0.485 | 0.00% | 823 | 8.949 | 0.00% |
| 10_5_04 | 776 | 776 | 0.297 | 0.00% | 776 | 6.458 | 0.00% |
| 10_5_05 | 798 | 798 | 0.313 | 0.00% | 798 | 4.176 | 0.00% |
| 10_5_06 | 849 | 849 | 0.281 | 0.00% | 849 | 5.695 | 0.00% |
| 10_5_07 | 843 | 843 | 0.610 | 0.00% | 843 | 14.168 | 0.00% |
| 10_5_08 | 768 | 768 | 0.406 | 0.00% | 768 | 9.167 | 0.00% |
| 10_5_09 | 841 | 841 | 0.250 | 0.00% | 841 | 2.803 | 0.00% |
| 10_5_10 | 719 | 719 | 0.375 | 0.00% | 719 | 5.311 | 0.00% |
| **10_5_Average** | **793.60** | **793.60** | **0.427** | **0.00%** | **793.60** | **7.179** | **0.00%** |
| 20_5_01 | 1414 | 1454 | 3600 | 43.19% | 1414 | 3600 | 19.80% |
| 20_5_02 | 1481 | 1489 | 3600 | 45.06% | 1481 | 3600 | 13.44% |
| 20_5_03 | 1588 | 1591 | 3600 | 45.82% | 1588 | 3600 | 19.27% |
| 20_5_04 | 1355 | 1385 | 3600 | 38.63% | 1355 | 3600 | 19.70% |
| 20_5_05 | 1520 | 1537 | 3600 | 43.40% | 1520 | 3600 | 13.09% |
| 20_5_06 | 1333 | 1338 | 3600 | 42.38% | 1333 | 3600 | 21.91% |
| 20_5_07 | 1388 | 1401 | 3600 | 37.19% | 1388 | 3600 | 18.23% |
| 20_5_08 | 1340 | 1341 | 3600 | 39.90% | 1346 | 3600 | 19.99% |
| 20_5_09 | 1499 | 1503 | 3600 | 38.12% | 1505 | 3600 | 12.49% |
| 20_5_10 | 1546 | 1564 | 3600 | 43.99% | 1560 | 3600 | 21.35% |
| **20_5_Average** | **1446.40** | **1460.30** | **3600** | **41.77%** | **1449.00** | **3600** | **17.93%** |
| 30_5_01 | 2072 | 2250 | 3600 | 69.39% | 2083 | 3600 | 13.54% |
| 30_5_02 | 1960 | 2073 | 3600 | 66.47% | 1968 | 3600 | 21.34% |
| 30_5_03 | 2029 | 2169 | 3600 | 68.93% | 2044 | 3600 | 20.94% |
| 30_5_04 | 2111 | 2307 | 3600 | 67.58% | 2126 | 3600 | 16.32% |
| 30_5_05 | 1967 | 2126 | 3600 | 68.67% | 1979 | 3600 | 13.64% |
| 30_5_06 | 2127 | 2275 | 3600 | 67.78% | 2130 | 3600 | 13.43% |
| 30_5_07 | 2036 | 2183 | 3600 | 66.41% | 2041 | 3600 | 15.24% |
| 30_5_08 | 2051 | 2141 | 3600 | 65.34% | 2052 | 3600 | 17.79% |
| 30_5_09 | 2046 | 2234 | 3600 | 68.89% | 2057 | 3600 | 16.48% |
| 30_5_10 | 1546 | 2197 | 3600 | 68.27% | 2058 | 3600 | 20.46% |
| **30_5_Average** | **1994,50** | **2195.50** | **3600** | **67.77%** | **2053.80** | **3600** | **16.92%** |

## Table A1.1.(Cont'd.) MILP and CP Comparison Table for 5 machines (VRF Instances)

| Instance | Opt. | MILP | Time (Seconds) | Gap % | CP | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|---|
| 40_5_01 | 2842 | 3034 | 3600 | 75.54% | 2866 | 3600 | 17.48% |
| 40_5_02 | 2875 | 3187 | 3600 | 79.48% | 2910 | 3600 | 17.25% |
| 40_5_03 | 2592 | 2847 | 3600 | 78.44% | 2607 | 3600 | 16.46% |
| 40_5_04 | 2637 | 2969 | 3600 | 78.71% | 2661 | 3600 | 19.24% |
| 40_5_05 | 2738 | 3116 | 3600 | 79.14% | 2761 | 3600 | 19.09% |
| 40_5_06 | 2598 | 2933 | 3600 | 77.22% | 2617 | 3600 | 17.69% |
| 40_5_07 | 2649 | 3043 | 3600 | 79.20% | 2672 | 3600 | 18.45% |
| 40_5_08 | 2829 | 3155 | 3600 | 77.97% | 2865 | 3600 | 15.92% |
| 40_5_09 | 2753 | 3164 | 3600 | 79.11% | 2770 | 3600 | 16.97% |
| 40_5_10 | 2797 | 3145 | 3600 | 77.62% | 2817 | 3600 | 16.65% |
| **40_5_Average** | **2731.00** | **3059.30** | **3600** | **78.24%** | **2754.60** | **3600** | **17.52%** |
| 50_5_01 | 3577 | 3991 | 3600 | 82.24% | 3614 | 3600 | 16.02% |
| 50_5_02 | 3303 | 3738 | 3600 | 82.00% | 3324 | 3600 | 14.17% |
| 50_5_03 | 3289 | 3867 | 3600 | 85.00% | 3327 | 3600 | 17.43% |
| 50_5_04 | 3391 | 3774 | 3600 | 82.94% | 3404 | 3600 | 17.33% |
| 50_5_05 | 3405 | 3853 | 3600 | 80.06% | 3430 | 3600 | 16.44% |
| 50_5_06 | 3302 | 3946 | 3600 | 83.86% | 3325 | 3600 | 14.56% |
| 50_5_07 | 3088 | 3630 | 3600 | 82.42% | 3115 | 3600 | 16.53% |
| 50_5_08 | 3238 | 3796 | 3600 | 81.88% | 3315 | 3600 | 19.03% |
| 50_5_09 | 3117 | 3793 | 3600 | 84.71% | 3178 | 3600 | 17.62% |
| 50_5_10 | 3372 | 3779 | 3600 | 80.89% | 3398 | 3600 | 16.60% |
| **50_5_Average** | **3308.20** | **3816.70** | **3600** | **82.60%** | **3343.00** | **3600** | **16.57%** |
| 60_5_01 | 3906 | 4725 | 3600 | 85.95% | 4004 | 3600 | 16.33% |
| 60_5_02 | 3779 | 4471 | 3600 | 85.83% | 3830 | 3600 | 20.73% |
| 60_5_03 | 3858 | 4583 | 3600 | 87.56% | 3904 | 3600 | 18.24% |
| 60_5_04 | 3899 | 4683 | 3600 | 84.97% | 3982 | 3600 | 17.98% |
| 60_5_05 | 3941 | 4616 | 3600 | 84.29% | 4014 | 3600 | 20.50% |
| 60_5_06 | 3758 | 4319 | 3600 | 85.14% | 3822 | 3600 | 18.60% |
| 60_5_07 | 4001 | 4610 | 3600 | 85.55% | 4048 | 3600 | 18.11% |
| 60_5_08 | 4138 | 4863 | 3600 | 84.60% | 4210 | 3600 | 18.86% |
| 60_5_09 | 3784 | 4481 | 3600 | 85.11% | 3825 | 3600 | 19.01% |
| 60_5_10 | 3980 | 4647 | 3600 | 95.52% | 4007 | 3600 | 8.58% |
| **60_5_Average** | **3904.40** | **4599.80** | **3600** | **86.45%** | **3964.60** | **3600** | **17.69%** |

## Table A.2. MILP and CP Comparison Table for 10 machines (VRF Instances)

| Instance | Opt. | MILP | Time (Seconds) | Gap % | CP | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|---|
| 10_10_01 | 1253 | 1253 | 0.360 | 0.00% | 1253 | 16.310 | 0.00% |
| 10_10_02 | 1278 | 1278 | 0.641 | 0.00% | 1278 | 39.157 | 0.00% |
| 10_10_03 | 1171 | 1171 | 0.172 | 0.00% | 1171 | 7.034 | 0.00% |
| 10_10_04 | 1181 | 1181 | 0.297 | 0.00% | 1181 | 15.634 | 0.00% |
| 10_10_05 | 1294 | 1294 | 1.187 | 0.00% | 1294 | 43.662 | 0.00% |
| 10_10_06 | 1198 | 1198 | 0.313 | 0.00% | 1198 | 24.013 | 0.00% |
| 10_10_07 | 1256 | 1256 | 0.391 | 0.00% | 1256 | 31.632 | 0.00% |
| 10_10_08 | 1220 | 1220 | 0.250 | 0.00% | 1220 | 18.071 | 0.00% |
| 10_10_09 | 1243 | 1243 | 0.359 | 0.00% | 1243 | 25.590 | 0.00% |
| 10_10_10 | 1317 | 1317 | 0.797 | 0.00% | 1317 | 26.374 | 0.00% |
| **10_10_Average** | **1241.10** | **1241.10** | **0.477** | **0.00%** | **1241.10** | **24.748** | **0.00%** |
| 20_10_01 | 2017 | 2036 | 3600 | 35.71% | 2017 | 3600 | 31.98% |
| 20_10_02 | 1998 | 2022 | 3600 | 36.10% | 1998 | 3600 | 28.13% |
| 20_10_03 | 2036 | 2036 | 3600 | 33.79% | 2045 | 3600 | 28.02% |
| 20_10_04 | 1932 | 1965 | 3600 | 38.07% | 1952 | 3600 | 32.94% |
| 20_10_05 | 2032 | 2051 | 3600 | 35.15% | 2032 | 3600 | 29.08% |
| 20_10_06 | 2059 | 2133 | 3600 | 41.91% | 2059 | 3600 | 30.45% |
| 20_10_07 | 2051 | 2069 | 3600 | 35.86% | 2051 | 3600 | 30.47% |
| 20_10_08 | 2018 | 2023 | 3600 | 37.47% | 2018 | 3600 | 28.59% |
| 20_10_09 | 1979 | 1996 | 3600 | 32.62% | 1990 | 3600 | 30.05% |
| 20_10_10 | 1963 | 1984 | 3600 | 35.33% | 1965 | 3600 | 27.84% |
| **20_10_Average** | **2008.50** | **2031.50** | **3600** | **36.20%** | **2012.70** | **3600** | **29,76%** |
| 30_10_01 | 2653 | 2918 | 3600 | 63.91% | 2689 | 3600 | 32.21% |
| 30_10_02 | 2861 | 2971 | 3600 | 60.85% | 2897 | 3600 | 32.72% |
| 30_10_03 | 2796 | 3031 | 3600 | 64.09% | 2801 | 3600 | 29.74% |
| 30_10_04 | 2762 | 2995 | 3600 | 60.43% | 2801 | 3600 | 35.34% |
| 30_10_05 | 2773 | 3026 | 3600 | 64.67% | 2786 | 3600 | 29.61% |
| 30_10_06 | 2808 | 3179 | 3600 | 63.16% | 2822 | 3600 | 30.62% |
| 30_10_07 | 2683 | 2937 | 3600 | 63.36% | 2708 | 3600 | 32.64% |
| 30_10_08 | 2532 | 2643 | 3600 | 62.32% | 2569 | 3600 | 30.40% |
| 30_10_09 | 2693 | 2804 | 3600 | 59.52% | 2705 | 3600 | 33.42% |
| 30_10_10 | 2647 | 2821 | 3600 | 61.60% | 2670 | 3600 | 31.39% |
| **30_10_Average** | **2720.80** | **2932.50** | **3600** | **62.39%** | **2744.80** | **3600** | **31.81%** |

**Table A.2.(Cont'd.) MILP and CP Comparison Table for 10 machines (VRF Instances)**

| Instance | Opt. | MILP | Time (Seconds) | Gap % | CP | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|---|
| 40_10_01 | 3550 | 4292 | 3600 | 75.33% | 3563 | 3600 | 34.24% |
| 40_10_02 | 3416 | 3881 | 3600 | 70.94% | 3416 | 3600 | 31.70% |
| 40_10_03 | 3408 | 3944 | 3600 | 76.17% | 3494 | 3600 | 33.60% |
| 40_10_04 | 3622 | 4184 | 3600 | 74.09% | 3704 | 3600 | 35.39% |
| 40_10_05 | 3488 | 4045 | 3600 | 75.87% | 3549 | 3600 | 33.90% |
| 40_10_06 | 3565 | 4014 | 3600 | 74.56% | 3599 | 3600 | 31.95% |
| 40_10_07 | 3496 | 4028 | 3600 | 76.48% | 3519 | 3600 | 31.77% |
| 40_10_08 | 3427 | 4004 | 3600 | 75.75% | 3467 | 3600 | 31.04% |
| 40_10_09 | 3501 | 3894 | 3600 | 73.47% | 3567 | 3600 | 35.13% |
| 40_10_10 | 3447 | 3937 | 3600 | 74.02% | 3451 | 3600 | 29.67% |
| **40_10_Average** | **3492.00** | **4022.30** | **3600** | **74.67%** | **3532.90** | **3600** | **32.84%** |
| 50_10_01 | 4121 | 4723 | 3600 | 79.32% | 4179 | 3600 | 32.40% |
| 50_10_02 | 4261 | 5067 | 3600 | 79.18% | 4388 | 3600 | 33.20% |
| 50_10_03 | 4227 | 5014 | 3600 | 78.40% | 4299 | 3600 | 31.71% |
| 50_10_04 | 4320 | 4873 | 3600 | 78.49% | 4360 | 3600 | 33.12% |
| 50_10_05 | 4356 | 5093 | 3600 | 78.55% | 4423 | 3600 | 27.65% |
| 50_10_06 | 4205 | 4912 | 3600 | 80.34% | 4248 | 3600 | 27.00% |
| 50_10_07 | 4096 | 4715 | 3600 | 80.78% | 4215 | 3600 | 33.45% |
| 50_10_08 | 4322 | 5015 | 3600 | 79.30% | 4365 | 3600 | 31.59% |
| 50_10_09 | 4289 | 5103 | 3600 | 80.23% | 4367 | 3600 | 32.06% |
| 50_10_10 | 4268 | 4819 | 3600 | 80.22% | 4376 | 3600 | 32.52% |
| **50_10_Average** | **4246.50** | **4933.40** | **3600** | **79.48%** | **4322.00** | **3600** | **31.47%** |
| 60_10_01 | 5067 | 5980 | 3600 | 83.34% | 5175 | 3600 | 35.03% |
| 60_10_02 | 5185 | 6547 | 3600 | 82.89% | 5271 | 3600 | 31.89% |
| 60_10_03 | 4953 | 6101 | 3600 | 82.54% | 5095 | 3600 | 33.72% |
| 60_10_04 | 5006 | 6049 | 3600 | 83.20% | 5054 | 3600 | 32.63% |
| 60_10_05 | 5140 | 6241 | 3600 | 82.10% | 5260 | 3600 | 34.24% |
| 60_10_06 | 5146 | 6222 | 3600 | 84.58% | 5226 | 3600 | 32.24% |
| 60_10_07 | 5130 | 6354 | 3600 | 82.67% | 5218 | 3600 | 32.06% |
| 60_10_08 | 4976 | 6083 | 3600 | 83.45% | 5065 | 3600 | 30.52% |
| 60_10_09 | 5001 | 6129 | 3600 | 84.29% | 5129 | 3600 | 29.48% |
| 60_10_10 | 5040 | 6059 | 3600 | 82.14% | 5111 | 3600 | 32.03% |
| **60_10_Average** | **5064.40** | **6176.50** | **3600** | **83.12%** | **5160.40** | **3600** | **32.38%** |

## Table A.3. MILP and CP Comparison Table for 15 machines (VRF Instances)

| Instance | Opt. | MILP | Time (Seconds) | Gap % | CP | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|---|
| 10_15_01 | 1516 | 1516 | 0.312 | 0.00% | 1516 | 45.260 | 0.00% |
| 10_15_02 | 1596 | 1596 | 0.281 | 0.00% | 1596 | 32.278 | 0.00% |
| 10_15_03 | 1611 | 1611 | 0.297 | 0.00% | 1611 | 46.715 | 0.00% |
| 10_15_04 | 1649 | 1649 | 0.406 | 0.00% | 1649 | 65.833 | 0.00% |
| 10_15_05 | 1602 | 1602 | 0.578 | 0.00% | 1602 | 49.938 | 0.00% |
| 10_15_06 | 1529 | 1529 | 0.234 | 0.00% | 1529 | 48.409 | 0.00% |
| 10_15_07 | 1702 | 1702 | 0.500 | 0.00% | 1702 | 65.594 | 0.00% |
| 10_15_08 | 1720 | 1720 | 0.265 | 0.00% | 1720 | 54.795 | 0.00% |
| 10_15_09 | 1683 | 1683 | 0.422 | 0.00% | 1683 | 62.265 | 0.00% |
| 10_15_10 | 1687 | 1687 | 0.500 | 0.00% | 1687 | 50.821 | 0.00% |
| **10_15_Average** | **1629.50** | **1629.50** | **0.380** | **0.00%** | **1629.50** | **52,19** | **0.00%** |
| 20_15_01 | 2663 | 2703 | 3600 | 33.26% | 2663 | 3600 | 36.99% |
| 20_15_02 | 2523 | 2553 | 3600 | 30.36% | 2523 | 3600 | 35.71% |
| 20_15_03 | 2392 | 2392 | 3600 | 32.48% | 2392 | 3600 | 34.91% |
| 20_15_04 | 2392 | 2417 | 3600 | 36.08% | 2392 | 3600 | 33.90% |
| 20_15_05 | 2502 | 2509 | 3600 | 34.32% | 2503 | 3600 | 31.48% |
| 20_15_06 | 2634 | 2634 | 3600 | 35.80% | 2634 | 3600 | 36.18% |
| 20_15_07 | 2580 | 2642 | 3600 | 31.26% | 2580 | 3600 | 33.29% |
| 20_15_08 | 2521 | 2545 | 3600 | 34.03% | 2531 | 3600 | 37.97% |
| 20_15_09 | 2511 | 2513 | 3600 | 31.24% | 2520 | 3600 | 31.31% |
| 20_15_10 | 2519 | 2561 | 3600 | 37.80% | 2519 | 3600 | 33.78% |
| **20_15_Average** | **2523.70** | **2546.90** | **3600** | **33.66%** | **2525.70** | **3600** | **34.55%** |
| 30_15_01 | 3347 | 3634 | 3600 | 62.33% | 3374 | 3600 | 36.93% |
| 30_15_02 | 3243 | 3671 | 3600 | 58.91% | 3243 | 3600 | 35.71% |
| 30_15_03 | 3301 | 3520 | 3600 | 58.52% | 3302 | 3600 | 37.49% |
| 30_15_04 | 3406 | 3619 | 3600 | 60.49% | 3441 | 3600 | 34.90% |
| 30_15_05 | 3463 | 3765 | 3600 | 60.13% | 3502 | 3600 | 36.89% |
| 30_15_06 | 3478 | 3765 | 3600 | 60.13% | 3502 | 3600 | 36.89% |
| 30_15_07 | 3416 | 3719 | 3600 | 58.78% | 3543 | 3600 | 41.77% |
| 30_15_08 | 3444 | 3605 | 3600 | 59.81% | 3461 | 3600 | 38.40% |
| 30_15_09 | 3314 | 3761 | 3600 | 61.85% | 3470 | 3600 | 38.65% |
| 30_15_10 | 3390 | 3557 | 3600 | 58.20% | 3360 | 3600 | 39.94% |
| **30_15_Average** | **3380.20** | **3661.60** | **3600** | **59.91%** | **3419.80** | **3600** | **37.76%** |

**Table A.3.(Cont'd.) MILP and CP Comparison Table for 15 machines (VRF Instances)**

| Instance | Opt. | MILP | Time (Seconds) | Gap % | CP | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|---|
| 40_15_01 | 4370 | 5038 | 3600 | 72.77% | 4388 | 3600 | 37.08% |
| 40_15_02 | 4214 | 4828 | 3600 | 71.06% | 4311 | 3600 | 39.20% |
| 40_15_03 | 4251 | 5019 | 3600 | 73.58% | 4271 | 3600 | 36.29% |
| 40_15_04 | 4249 | 4845 | 3600 | 73.13% | 4358 | 3600 | 39.08% |
| 40_15_05 | 4353 | 4682 | 3600 | 70.61% | 4458 | 3600 | 39.26% |
| 40_15_06 | 4120 | 4626 | 3600 | 69.93% | 4132 | 3600 | 37.66% |
| 40_15_07 | 4299 | 4946 | 3600 | 71.96% | 4374 | 3600 | 39.32% |
| 40_15_08 | 4279 | 4923 | 3600 | 70.81% | 4346 | 3600 | 38.33% |
| 40_15_09 | 4116 | 4628 | 3600 | 70.46% | 4139 | 3600 | 39.07% |
| 40_15_10 | 4301 | 4783 | 3600 | 71.94% | 4341 | 3600 | 38.10% |
| **40_15_Average** | **4255.20** | **4831.80** | **3600** | **71.63%** | **4311.80** | **3600** | **38.34%** |
| 50_15_01 | 4972 | 5768 | 3600 | 78.54% | 5042 | 3600 | 39.79% |
| 50_15_02 | 5079 | 5822 | 3600 | 75.95% | 5160 | 3600 | 38.41% |
| 50_15_03 | 5136 | 5993 | 3600 | 77.79% | 5255 | 3600 | 40.67% |
| 50_15_04 | 5248 | 6176 | 3600 | 76.89% | 5434 | 3600 | 38.35% |
| 50_15_05 | 5092 | 6097 | 3600 | 78.22% | 5212 | 3600 | 39.87% |
| 50_15_06 | 5194 | 6078 | 3600 | 77.26% | 5280 | 3600 | 39.87% |
| 50_15_07 | 5297 | 6069 | 3600 | 76.57% | 5360 | 3600 | 38.45% |
| 50_15_08 | 5174 | 6188 | 3600 | 77.84% | 5312 | 3600 | 38.99% |
| 50_15_09 | 5096 | 6205 | 3600 | 78.69% | 5132 | 3600 | 41.11% |
| 50_15_10 | 5173 | 6018 | 3600 | 78.58% | 5223 | 3600 | 38.75% |
| **50_15_Average** | **5146.10** | **6041.40** | **3600** | **77.63%** | **5241.00** | **3600** | **39.43%** |
| 60_15_01 | 5972 | 7080 | 3600 | 81.69% | 6192 | 3600 | 39.11% |
| 60_15_02 | 5965 | 7252 | 3600 | 81.54% | 6061 | 3600 | 39.50% |
| 60_15_03 | 6070 | 7163 | 3600 | 80.09% | 6260 | 3600 | 41.34% |
| 60_15_04 | 5974 | 7233 | 3600 | 80.15% | 6081 | 3600 | 41.82% |
| 60_15_05 | 6004 | 7306 | 3600 | 82.53% | 6096 | 3600 | 40.16% |
| 60_15_06 | 6149 | 7614 | 3600 | 81.89% | 6321 | 3600 | 41.53% |
| 60_15_07 | 6059 | 7131 | 3600 | 81.24% | 6226 | 3600 | 42.27% |
| 60_15_08 | 5974 | 7236 | 3600 | 81.41% | 6053 | 3600 | 41.20% |
| 60_15_09 | 5760 | 6929 | 3600 | 81.28% | 5939 | 3600 | 39.20% |
| 60_15_10 | 6092 | 7636 | 3600 | 81.05% | 6206 | 3600 | 40.06% |
| **60_15_Average** | **6001.90** | **7258.00** | **3600** | **81.29%** | **6143.50** | **3600** | **40.62%** |

## Table A.4. MILP and CP Comparison Table for 20 machines (VRF Instances)

| Instance | Opt. | MILP | Time (Seconds) | Gap % | CP | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|---|
| 10_20_01 | 1913 | 1913 | 0.282 | 0.00% | 1913 | 119.638 | 0.00% |
| 10_20_02 | 1973 | 1973 | 0.156 | 0.00% | 1973 | 78.135 | 0.00% |
| 10_20_03 | 1989 | 1989 | 0.594 | 0.00% | 1989 | 118.302 | 0.00% |
| 10_20_04 | 1971 | 1971 | 0.594 | 0.00% | 1971 | 99.448 | 0.00% |
| 10_20_05 | 1979 | 1979 | 0.657 | 0.00% | 1979 | 242.102 | 0.00% |
| 10_20_06 | 2152 | 2152 | 0.297 | 0.00% | 2152 | 94.656 | 0.00% |
| 10_20_07 | 1893 | 1893 | 0.187 | 0.00% | 1893 | 60.239 | 0.00% |
| 10_20_08 | 1933 | 1933 | 0.485 | 0.00% | 1933 | 139.517 | 0.00% |
| 10_20_09 | 1941 | 1941 | 0.203 | 0.00% | 1941 | 74.479 | 0.00% |
| 10_20_10 | 1876 | 1876 | 0.343 | 0.00% | 1876 | 70.311 | 0.00% |
| **10_20_Average** | **1962.00** | **1962.00** | **0.380** | **0.00%** | **1962.00** | **109.683** | **0.00%** |
| 20_20_01 | 3082 | 3085 | 3600 | 28.07% | 3104 | 3600 | 39,66% |
| 20_20_02 | 2872 | 2896 | 3600 | 30.52% | 2872 | 3600 | 35,52% |
| 20_20_03 | 2935 | 2997 | 3600 | 28.09% | 2935 | 3600 | 36,08% |
| 20_20_04 | 2828 | 2900 | 3600 | 30.93% | 2828 | 3600 | 37,02% |
| 20_20_05 | 3078 | 3145 | 3600 | 28.20% | 3078 | 3600 | 39,90% |
| 20_20_06 | 3172 | 3202 | 3600 | 31.61% | 3174 | 3600 | 39,79% |
| 20_20_07 | 2999 | 3029 | 3600 | 22.15% | 2999 | 3600 | 37,75% |
| 20_20_08 | 2837 | 2853 | 3600 | 32.46% | 2837 | 3600 | 32,92% |
| 20_20_09 | 3094 | 3166 | 3600 | 31.14% | 3094 | 3600 | 34,91% |
| 20_20_10 | 2884 | 2912 | 3600 | 29.29% | 2884 | 3600 | 36,17% |
| **20_20_Average** | **2978.10** | **3018.50** | **3600** | **29.25%** | **2980.50** | **3600** | **37.29%** |
| 30_20_01 | 3894 | 4214 | 3600 | 56.12% | 3950 | 3600 | 43.59% |
| 30_20_02 | 4017 | 4320 | 3600 | 57.22% | 4112 | 3600 | 41.25% |
| 30_20_03 | 4022 | 4317 | 3600 | 59.44% | 4061 | 3600 | 40.09% |
| 30_20_04 | 3786 | 4174 | 3600 | 55.74% | 3823 | 3600 | 39.34% |
| 30_20_05 | 3781 | 4016 | 3600 | 56.05% | 3785 | 3600 | 39.00% |
| 30_20_06 | 3971 | 4143 | 3600 | 58.39% | 3991 | 3600 | 40.62% |
| 30_20_07 | 3999 | 4324 | 3600 | 56.68% | 4083 | 3600 | 41.07% |
| 30_20_08 | 4016 | 4322 | 3600 | 57.33% | 4079 | 3600 | 39.25% |
| 30_20_09 | 4019 | 4312 | 3600 | 56.98% | 4049 | 3600 | 39.89% |
| 30_20_10 | 4113 | 4273 | 3600 | 55.51% | 4145 | 3600 | 41.64% |
| **30_20_Average** | **3961.80** | **4241.50** | **3600** | **56.95%** | **4007.80** | **3600** | **40.57%** |

**Table A.4.(Cont'd.) MILP and CP Comparison Table for 20 machines (VRF Instances)**

| Instance | Opt. | MILP | Time (Seconds) | Gap % | CP | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|---|
| 40_20_01 | 4935 | 5519 | 3600 | 69.14% | 5029 | 3600 | 39.13% |
| 40_20_02 | 4854 | 5605 | 3600 | 68.22% | 4932 | 3600 | 42.98% |
| 40_20_03 | 5103 | 5753 | 3600 | 69.42% | 5198 | 3600 | 46.11% |
| 40_20_04 | 4837 | 5847 | 3600 | 70.40% | 4905 | 3600 | 41.33% |
| 40_20_05 | 4712 | 5489 | 3600 | 68.76% | 4837 | 3600 | 43.83% |
| 40_20_06 | 4936 | 5478 | 3600 | 68.18% | 4969 | 3600 | 43.87% |
| 40_20_07 | 5092 | 5665 | 3600 | 67.84% | 5107 | 3600 | 44.25% |
| 40_20_08 | 4999 | 5334 | 3600 | 66.65% | 5067 | 3600 | 43.02% |
| 40_20_09 | 5041 | 5664 | 3600 | 67.80% | 5154 | 3600 | 43.07% |
| 40_20_10 | 4726 | 5127 | 3600 | 69.09% | 4732 | 3600 | 40.87% |
| **40_20_Average** | **4923.50** | **5548.10** | **3600** | **68.55%** | **4993.00** | **3600** | **42.85%** |
| 50_20_01 | 5854 | 6869 | 3600 | 74.76% | 6084 | 3600 | 46.01% |
| 50_20_02 | 5825 | 6633 | 3600 | 74.36% | 5973 | 3600 | 43.85% |
| 50_20_03 | 5952 | 7024 | 3600 | 74.75% | 6077 | 3600 | 43.11% |
| 50_20_04 | 5960 | 6954 | 3600 | 73.35% | 6000 | 3600 | 44.57% |
| 50_20_05 | 5893 | 6957 | 3600 | 74.72% | 5988 | 3600 | 44.66% |
| 50_20_06 | 6042 | 7067 | 3600 | 74.13% | 6175 | 3600 | 44.87% |
| 50_20_07 | 5984 | 7146 | 3600 | 77.18% | 6112 | 3600 | 44.70% |
| 50_20_08 | 5906 | 7000 | 3600 | 75.91% | 6011 | 3600 | 43.19% |
| 50_20_09 | 5977 | 6859 | 3600 | 76.02% | 6134 | 3600 | 43.95% |
| 50_20_10 | 5926 | 7241 | 3600 | 76.85% | 6048 | 3600 | 43.70% |
| **50_20_Average** | **5931.90** | **6975.00** | **3600** | **75.20%** | **6060.20** | **3600** | **44.26%** |
| 60_20_01 | 6925 | 8615 | 3600 | 79.34% | 7090 | 3600 | 45.37% |
| 60_20_02 | 6928 | 8369 | 3600 | 79.01% | 7011 | 3600 | 44.49% |
| 60_20_03 | 7151 | 8791 | 3600 | 80.29% | 7409 | 3600 | 46.19% |
| 60_20_04 | 7077 | 8867 | 3600 | 80.62% | 7222 | 3600 | 45.04% |
| 60_20_05 | 6699 | 8014 | 3600 | 78.77% | 6848 | 3600 | 43.93% |
| 60_20_06 | 6781 | 8166 | 3600 | 78.32% | 6973 | 3600 | 44.07% |
| 60_20_07 | 6909 | 8355 | 3600 | 78.90% | 7061 | 3600 | 43.76% |
| 60_20_08 | 6871 | 8234 | 3600 | 78.36% | 7071 | 3600 | 45.28% |
| 60_20_09 | 6833 | 8152 | 3600 | 78.69% | 7035 | 3600 | 44.36% |
| 60_20_10 | 6724 | 8262 | 3600 | 79.33% | 6962 | 3600 | 44.30% |
| **60_20_Average** | **6889.80** | **8382.50** | **3600** | **79.16%** | **7068.20** | **3600** | **44.68%** |

## Table A.5. MILP-Prime and CP-Prime Comparison Table with Lower and Upper Bounds for 5 machines (VRF Instances)

| Instance | LB | MILP-Prime | Time (Seconds) | Gap % | CP-Prime | Time (Seconds) | Gap % | UB |
|---|---|---|---|---|---|---|---|---|
| 10_5_01 | 442 | 760 | 1,281 | 0.00% | 760 | 10.493 | 0.00% | 2052 |
| 10_5_02 | 458 | 759 | 0,438 | 0.00% | 759 | 4.502 | 0.00% | 2158 |
| 10_5_03 | 351 | 823 | 0,437 | 0.00% | 823 | 7.742 | 0.00% | 2296 |
| 10_5_04 | 468 | 776 | 0,313 | 0.00% | 776 | 6.524 | 0.00% | 2243 |
| 10_5_05 | 493 | 798 | 0,422 | 0.00% | 798 | 3.933 | 0.00% | 2371 |
| 10_5_06 | 496 | 849 | 0,656 | 0.00% | 849 | 6.396 | 0.00% | 2466 |
| 10_5_07 | 482 | 843 | 1,141 | 0.00% | 843 | 14.193 | 0.00% | 2380 |
| 10_5_08 | 415 | 768 | 0,484 | 0.00% | 768 | 10.551 | 0.00% | 2132 |
| 10_5_09 | 546 | 841 | 0,375 | 0.00% | 841 | 2.419 | 0.00% | 2678 |
| 10_5_10 | 442 | 719 | 0,735 | 0.00% | 719 | 8.889 | 0.00% | 2015 |
| **10_5_Average** | **459.30** | **793.60** | **0.628** | **0.00%** | **793.60** | **7.564** | **0.00%** | **2279.10** |
| 20_5_01 | 1022 | 1417 | 3600 | 22,79% | 1414 | 3600 | 19.80% | 483 |
| 20_5_02 | 1074 | 1527 | 3600 | 26,26% | 1481 | 3600 | 13.44% | 5262 |
| 20_5_03 | 967 | 1611 | 3600 | 35,44% | 1588 | 3600 | 19.27% | 5298 |
| 20_5_04 | 937 | 1373 | 3600 | 26,58% | 1355 | 3600 | 19.70% | 4713 |
| 20_5_05 | 1094 | 1523 | 3600 | 22,78% | 1524 | 3600 | 13.32% | 5394 |
| 20_5_06 | 926 | 1346 | 3600 | 27,34% | 1336 | 3600 | 22.08% | 4647 |
| 20_5_07 | 987 | 1391 | 3600 | 23,79% | 1393 | 3600 | 18.52% | 4897 |
| 20_5_08 | 887 | 1340 | 3600 | 29,70% | 1343 | 3600 | 19.81% | 4732 |
| 20_5_09 | 1025 | 1536 | 3600 | 28,90% | 1505 | 3600 | 12.49% | 5109 |
| 20_5_10 | 1038 | 1603 | 3600 | 30,63% | 1546 | 3600 | 20.63% | 5294 |
| **20_5_Average** | **995.70** | **1466.70** | **3600** | **27.42%** | **1448,50** | **3600** | **17.91%** | **5019** |
| 30_5_01 | 1281 | 2329 | 3600 | 40.31% | 2082 | 3600 | 13.50% | 7695 |
| 30_5_02 | 1434 | 2117 | 3600 | 28.86% | 1974 | 3600 | 21.58% | 6913 |
| 30_5_03 | 1515 | 2180 | 3600 | 28.21% | 2058 | 3600 | 21.48% | 7466 |
| 30_5_04 | 1373 | 2305 | 3600 | 37.22% | 2127 | 3600 | 16.36% | 7631 |
| 30_5_05 | 1408 | 2181 | 3600 | 32.18% | 1984 | 3600 | 13.86% | 7298 |
| 30_5_06 | 1743 | 2307 | 3600 | 21.75% | 2127 | 3600 | 13.31% | 7434 |
| 30_5_07 | 1249 | 2185 | 3600 | 41.05% | 2041 | 3600 | 15.24% | 7085 |
| 30_5_08 | 1463 | 2169 | 3600 | 28.44% | 2062 | 3600 | 18.19% | 7737 |
| 30_5_09 | 1592 | 2215 | 3600 | 24.96% | 2046 | 3600 | 16.03% | 7475 |
| 30_5_10 | 1447 | 2267 | 3600 | 32.68% | 2056 | 3600 | 20.38% | 7325 |
| **30_5_Average** | **1450.50** | **2225.50** | **3600** | **31.57%** | **2055.70** | **3600** | **16.99%** | **7405.90** |

**Table A.5.(Cont'd.) MILP-Prime and CP-Prime Comparison Table with Lower and Upper Bounds for 5 machines (VRF Instances)**

| Instance | LB | MILP-Prime | Time (Seconds) | Gap % | CP-Prime | Time (Seconds) | Gap % | UB |
|---|---|---|---|---|---|---|---|---|
| 40_5_01 | 1868 | 3139 | 3600 | 38.22% | 2890 | 3600 | 18.17% | 10510 |
| 40_5_02 | 2111 | 3284 | 3600 | 33.55% | 2903 | 3600 | 17.05% | 10730 |
| 40_5_03 | 2022 | 2925 | 3600 | 30.32% | 2605 | 3600 | 16.39% | 9855 |
| 40_5_04 | 1966 | 3011 | 3600 | 33.94% | 2658 | 3600 | 19.15% | 9723 |
| 40_5_05 | 1991 | 3067 | 3600 | 33.64% | 2750 | 3600 | 18.76% | 10342 |
| 40_5_06 | 1862 | 2939 | 3600 | 35.62% | 2635 | 3600 | 18.25% | 9584 |
| 40_5_07 | 1976 | 2933 | 3600 | 29.86% | 2665 | 3600 | 18.24% | 10076 |
| 40_5_08 | 2114 | 3143 | 3600 | 30.86% | 2867 | 3600 | 15.97% | 10705 |
| 40_5_09 | 2168 | 3051 | 3600 | 26.71% | 2784 | 3600 | 17.39% | 10330 |
| 40_5_10 | 2050 | 3198 | 3600 | 35.39% | 2810 | 3600 | 16.44% | 10477 |
| **40_5_Average** | **2012.80** | **3069.00** | **3600** | **32.81%** | **2756.70** | **3600** | **17.58%** | **10233.20** |
| 50_5_01 | 2596 | 4128 | 3600 | 35.82% | 3632 | 3600 | 16.44% | 13656 |
| 50_5_02 | 2547 | 3746 | 3600 | 30.80% | 3334 | 3600 | 14.43% | 12549 |
| 50_5_03 | 2347 | 3868 | 3600 | 36.78% | 3342 | 3600 | 17.80% | 12343 |
| 50_5_04 | 2652 | 3968 | 3600 | 31.42% | 3424 | 3600 | 17.82% | 13017 |
| 50_5_05 | 2318 | 3974 | 3600 | 40.18% | 3439 | 3600 | 16.66% | 12751 |
| 50_5_06 | 2363 | 3927 | 3600 | 38.17% | 3343 | 3600 | 15.02% | 12349 |
| 50_5_07 | 2417 | 3692 | 3600 | 33.04% | 3127 | 3600 | 16.85% | 11847 |
| 50_5_08 | 2526 | 3822 | 3600 | 32.75% | 3287 | 3600 | 18.34% | 12202 |
| 50_5_09 | 2383 | 3599 | 3600 | 31.98% | 3136 | 3600 | 16.52% | 11801 |
| 50_5_10 | 2648 | 3968 | 3600 | 31.65% | 3423 | 3600 | 17.21% | 12983 |
| **50_5_Average** | **2479.70** | **3869.20** | **3600** | **34.26%** | **3348.70** | **3600** | **16.71%** | **12549.80** |
| 60_5_01 | 3114 | 4835 | 3600 | 34.76% | 3986 | 3600 | 15.96% | 15417 |
| 60_5_02 | 2863 | 4403 | 3600 | 34.11% | 3871 | 3600 | 21.57% | 14362 |
| 60_5_03 | 2780 | 4516 | 3600 | 36.91% | 3944 | 3600 | 19.07% | 14970 |
| 60_5_04 | 3114 | 4770 | 3600 | 33.48% | 3970 | 3600 | 17.73% | 15396 |
| 60_5_05 | 2960 | 4780 | 3600 | 37.44% | 3983 | 3600 | 19.88% | 14900 |
| 60_5_06 | 3005 | 4336 | 3600 | 29.61% | 3848 | 3600 | 19.15% | 14555 |
| 60_5_07 | 3127 | 4664 | 3600 | 32.26% | 4049 | 3600 | 18.13% | 15508 |
| 60_5_08 | 3311 | 4890 | 3600 | 30.92% | 4215 | 3600 | 18.96% | 15752 |
| 60_5_09 | 2964 | 4626 | 3600 | 34.71% | 3860 | 3600 | 19.74% | 14679 |
| 60_5_10 | 2572 | 4859 | 3600 | 46.07% | 4009 | 3600 | 8.63% | 15082 |
| **60_5_Average** | **2981.00** | **4667.90** | **3600** | **35.03%** | **3973.50** | **3600** | **17.88%** | **15062.10** |

## Table A.6. MILP-Prime and CP-Prime Comparison Table with Lower and Upper Bounds for 10 machines (VRF Instances)

| Instance | LB | MILP-Prime | Time (Seconds) | Gap % | CP-Prime | Time (Seconds) | Gap % | UB |
|---|---|---|---|---|---|---|---|---|
| 10_10_01 | 501 | 1253 | 0.578 | 0.00% | 1253 | 18.045 | 0.00% | 4776 |
| 10_10_02 | 535 | 1278 | 0.875 | 0.00% | 1278 | 38.839 | 0.00% | 4710 |
| 10_10_03 | 533 | 1171 | 0.485 | 0.00% | 1171 | 5.848 | 0.00% | 4791 |
| 10_10_04 | 433 | 1181 | 0.594 | 0.00% | 1181 | 13.020 | 0.00% | 4529 |
| 10_10_05 | 528 | 1294 | 2.781 | 0.00% | 1294 | 36.939 | 0.00% | 4899 |
| 10_10_06 | 522 | 1198 | 0.906 | 0.00% | 1198 | 21.141 | 0.00% | 4624 |
| 10_10_07 | 512 | 1256 | 0.843 | 0.00% | 1256 | 35.876 | 0.00% | 4733 |
| 10_10_08 | 530 | 1220 | 0.531 | 0.00% | 1220 | 16.817 | 0.00% | 4851 |
| 10_10_09 | 456 | 1243 | 0.469 | 0.00% | 1243 | 20.644 | 0.00% | 4675 |
| 10_10_10 | 353 | 1317 | 1.953 | 0.00% | 1317 | 30.815 | 0.00% | 4677 |
| **10_10_Average** | **490.30** | **1241.10** | **1.002** | **0.00%** | **1241.10** | **23.798** | **0.00%** | **4726.50** |
| 20_10_01 | 1037 | 2017 | 3600 | 36.77% | 2029 | 3600 | 32.38% | 9299 |
| 20_10_02 | 972 | 2017 | 3600 | 38.52% | 1998 | 3600 | 28.13% | 9811 |
| 20_10_03 | 1072 | 2046 | 3600 | 36.55% | 2045 | 3600 | 28.02% | 10207 |
| 20_10_04 | 926 | 1934 | 3600 | 36.29% | 1932 | 3600 | 32.25% | 9590 |
| 20_10_05 | 911 | 2045 | 3600 | 34.71% | 2032 | 3600 | 29.08% | 10285 |
| 20_10_06 | 1004 | 2127 | 3600 | 37.18% | 2059 | 3600 | 30.35% | 10193 |
| 20_10_07 | 1025 | 2089 | 3600 | 38.39% | 2051 | 3600 | 30.47% | 10293 |
| 20_10_08 | 848 | 2046 | 3600 | 41.30% | 2018 | 3600 | 28.59% | 9850 |
| 20_10_09 | 1054 | 1982 | 3600 | 34.35% | 1980 | 3600 | 29.70% | 10037 |
| 20_10_10 | 820 | 1975 | 3600 | 36.20% | 1965 | 3600 | 27.84% | 9856 |
| **20_10_Average** | **966.90** | **2027.80** | **3600** | **37.03%** | **2010.90** | **3600** | **29.68%** | **9942.10** |
| 30_10_01 | 1499 | 2879 | 3600 | 40.36% | 2682 | 3600 | 32.03% | 14132 |
| 30_10_02 | 1568 | 3091 | 3600 | 41.96% | 2867 | 3600 | 32.02% | 15465 |
| 30_10_03 | 1527 | 3009 | 3600 | 42.63% | 2839 | 3600 | 30.68% | 15153 |
| 30_10_04 | 1512 | 3053 | 3600 | 42.77% | 2762 | 3600 | 34.43% | 14527 |
| 30_10_05 | 1428 | 2998 | 3600 | 45.36% | 2793 | 3600 | 29.79% | 15314 |
| 30_10_06 | 1548 | 3092 | 3600 | 40.84% | 2828 | 3600 | 30.76% | 15039 |
| 30_10_07 | 1476 | 2899 | 3600 | 39.94% | 2704 | 3600 | 32.54% | 14651 |
| 30_10_08 | 1430 | 2720 | 3600 | 39.96% | 2572 | 3600 | 30.48% | 13910 |
| 30_10_09 | 1449 | 2899 | 3600 | 43.18% | 2703 | 3600 | 33.37% | 14196 |
| 30_10_10 | 1136 | 2826 | 3600 | 54.17% | 2661 | 3600 | 31.15% | 14058 |
| **30_10_Average** | **1457.30** | **2946.60** | **3600** | **43.12%** | **2741.10** | **3600** | **31.73%** | **14644.50** |

**Table A.6.(Cont'd.) MILP-Prime and CP-Prime Comparison Table with Lower and Upper Bounds for 10 machines (VRF Instances)**

| Instance | LB | MILP-Prime | Time (Seconds) | Gap % | CP-Prime | Time (Seconds) | Gap % | UB |
|---|---|---|---|---|---|---|---|---|
| 40_10_01 | 1929 | 4069 | 3600 | 47.84% | 3609 | 3600 | 35.08% | 19757 |
| 40_10_02 | 1894 | 3945 | 3600 | 46.86% | 3416 | 3600 | 31.70% | 19179 |
| 40_10_03 | 1893 | 3868 | 3600 | 44.62% | 3440 | 3600 | 32.56% | 19020 |
| 40_10_04 | 1979 | 4162 | 3600 | 46.66% | 3704 | 3600 | 35.39% | 19820 |
| 40_10_05 | 1724 | 3950 | 3600 | 51.29% | 3510 | 3600 | 33.16% | 19814 |
| 40_10_06 | 1935 | 4039 | 3600 | 46.44% | 3598 | 3600 | 31.93% | 20110 |
| 40_10_07 | 2044 | 4161 | 3600 | 44.89% | 3506 | 3600 | 31.52% | 20042 |
| 40_10_08 | 2054 | 4034 | 3600 | 42.95% | 3454 | 3600 | 30.78% | 19329 |
| 40_10_09 | 2016 | 4035 | 3600 | 43.91% | 3591 | 3600 | 35.56% | 19442 |
| 40_10_10 | 2014 | 4139 | 3600 | 46.46% | 3507 | 3600 | 30.80% | 19656 |
| **40_10_Average** | **1948.20** | **4040.20** | **3600** | **46.19%** | **3533.50** | **3600** | **32.85%** | **19616.90** |
| 50_10_01 | 2460 | 5106 | 3600 | 46.96% | 4195 | 3600 | 32.66% | 24410 |
| 50_10_02 | 2555 | 5012 | 3600 | 45.49% | 4326 | 3600 | 32.25% | 24205 |
| 50_10_03 | 2537 | 4921 | 3600 | 44.54% | 4366 | 3600 | 32.75% | 24466 |
| 50_10_04 | 2442 | 5189 | 3600 | 49.56% | 4414 | 3600 | 33.94% | 25278 |
| 50_10_05 | 2333 | 5094 | 3600 | 49.17% | 4420 | 3600 | 27.60% | 26021 |
| 50_10_06 | 2496 | 5170 | 3600 | 48.54% | 4265 | 3600 | 27.29% | 24782 |
| 50_10_07 | 2194 | 4911 | 3600 | 50.49% | 4118 | 3600 | 31.88% | 23647 |
| 50_10_08 | 2402 | 5124 | 3600 | 48.26% | 4385 | 3600 | 31.90% | 25678 |
| 50_10_09 | 2580 | 5118 | 3600 | 44.31% | 4402 | 3600 | 32.60% | 24720 |
| 50_10_10 | 2647 | 4976 | 3600 | 42.40% | 4343 | 3600 | 32.01% | 25042 |
| **50_10_Average** | **2464.60** | **5062.10** | **3600** | **46.97%** | **4323.40** | **3600** | **31.49%** | **24824.90** |
| 60_10_01 | 3040 | 6025 | 3600 | 47.10% | 5104 | 3600 | 34.13% | 29905 |
| 60_10_02 | 3267 | 6622 | 3600 | 48.24% | 5286 | 3600 | 32.08% | 30594 |
| 60_10_03 | 2998 | 6062 | 3600 | 48.13% | 5023 | 3600 | 32.77% | 29285 |
| 60_10_04 | 2733 | 6079 | 3600 | 51.81% | 5136 | 3600 | 33.70% | 29346 |
| 60_10_05 | 3100 | 6165 | 3600 | 46.06% | 5277 | 3600 | 34.45% | 30651 |
| 60_10_06 | 3106 | 6317 | 3600 | 47.17% | 5258 | 3600 | 32.66% | 30831 |
| 60_10_07 | 3211 | 6127 | 3600 | 44.03% | 5207 | 3600 | 31.92% | 31148 |
| 60_10_08 | 2943 | 6218 | 3600 | 49.53% | 5052 | 3600 | 30.34% | 29460 |
| 60_10_09 | 3304 | 6072 | 3600 | 42.09% | 5107 | 3600 | 29.18% | 29805 |
| 60_10_10 | 2803 | 6297 | 3600 | 53.32% | 5115 | 3600 | 32.08% | 30036 |
| **60_10_Average** | **3050.50** | **6198.40** | **3600** | **47.75%** | **5156.50** | **3600** | **32.33%** | **30106.10** |

## Table A.7. MILP-Prime and CP-Prime Comparison Table with Lower and Upper Bounds for 15 machines (VRF Instances)

| Instance | LB | MILP-Prime | Time (Seconds) | Gap % | CP-Prime | Time (Seconds) | Gap % | UB |
|---|---|---|---|---|---|---|---|---|
| 10_15_01 | 383 | 1516 | 0.984 | 0.00% | 1516 | 42.976 | 0.00% | 6562 |
| 10_15_02 | 492 | 1596 | 1.047 | 0.00% | 1596 | 29.139 | 0.00% | 7200 |
| 10_15_03 | 449 | 1611 | 1.141 | 0.00% | 1611 | 41.875 | 0.00% | 7079 |
| 10_15_04 | 455 | 1649 | 1.234 | 0.00% | 1649 | 67.427 | 0.00% | 7201 |
| 10_15_05 | 515 | 1602 | 2.687 | 0.00% | 1602 | 43.724 | 0.00% | 7151 |
| 10_15_06 | 450 | 1529 | 1.063 | 0.00% | 1529 | 44.335 | 0.00% | 7008 |
| 10_15_07 | 538 | 1702 | 3.343 | 0.00% | 1702 | 59.613 | 0.00% | 7006 |
| 10_15_08 | 541 | 1720 | 1.516 | 0.00% | 1720 | 45.920 | 0.00% | 7185 |
| 10_15_09 | 393 | 1683 | 1.875 | 0.00% | 1683 | 63.983 | 0.00% | 7395 |
| 10_15_10 | 432 | 1687 | 2437 | 0.00% | 1687 | 56.364 | 0.00% | 7684 |
| **10_15_Average** | **464.80** | **1629.50** | **1.730** | **0.00%** | **1629.50** | **49.540** | **0.00%** | **7147.10** |
| 20_15_01 | 1102 | 2663 | 3600 | 34.73% | 2663 | 3600 | 36.99% | 15332 |
| 20_15_02 | 970 | 2558 | 3600 | 33.73% | 2523 | 3600 | 35.71% | 14990 |
| 20_15_03 | 970 | 2405 | 3600 | 36.83% | 2392 | 3600 | 34.91% | 14418 |
| 20_15_04 | 873 | 2394 | 3600 | 36.17% | 2392 | 3600 | 33.90% | 14399 |
| 20_15_05 | 947 | 2523 | 3600 | 38.84% | 2503 | 3600 | 31.48% | 15289 |
| 20_15_06 | 1083 | 2634 | 3600 | 37.16% | 2634 | 3600 | 36.18% | 15114 |
| 20_15_07 | 888 | 2580 | 3600 | 32.59% | 2603 | 3600 | 33.88% | 14959 |
| 20_15_08 | 1015 | 2526 | 3600 | 35.27% | 2523 | 3600 | 37.77% | 14679 |
| 20_15_09 | 1100 | 2525 | 3600 | 29.38% | 2513 | 3600 | 31.12% | 14511 |
| 20_15_10 | 831 | 2553 | 3600 | 39.67% | 2519 | 3600 | 33.78% | 14860 |
| **20_15_Average** | **977.90** | **2536.10** | **3600** | **35.44%** | **2526.50** | **3600** | **34.57%** | **14855.10** |
| 30_15_01 | 1541 | 3761 | 3600 | 47.32% | 3360 | 3600 | 36.67% | 22112 |
| 30_15_02 | 1563 | 3662 | 3600 | 48.11% | 3258 | 3600 | 36.00% | 21353 |
| 30_15_03 | 1337 | 3549 | 3600 | 50.71% | 3324 | 3600 | 37.91% | 21587 |
| 30_15_04 | 1553 | 3736 | 3600 | 47.85% | 3406 | 3600 | 34.23% | 22827 |
| 30_15_05 | 1524 | 3807 | 3600 | 51.32% | 3520 | 3600 | 37.22% | 22650 |
| 30_15_06 | 1506 | 3796 | 3600 | 49.26% | 3482 | 3600 | 40.75% | 21928 |
| 30_15_07 | 1272 | 3555 | 3600 | 52.43% | 3447 | 3600 | 38.15% | 21889 |
| 30_15_08 | 1408 | 3672 | 3600 | 48.12% | 3479 | 3600 | 38.80% | 22705 |
| 30_15_09 | 1430 | 3553 | 3600 | 50.04% | 3329 | 3600 | 39.38% | 21002 |
| 30_15_10 | 1407 | 3618 | 3600 | 48.09% | 3431 | 3600 | 38.33% | 22381 |
| **30_15_Average** | **1454.10** | **3670.90** | **3600** | **49.33%** | **3403.60** | **3600** | **37.74%** | **22043.40** |

**Table A.7.(Cont'd.) MILP-Prime and CP-Prime Comparison Table with Lower and Upper Bounds for 15 machines (VRF Instances)**

| Instance | LB | MILP-Prime | Time (Seconds) | Gap % | CP-Prime | Time (Seconds) | Gap % | UB |
|---|---|---|---|---|---|---|---|---|
| 40_15_01 | 2106 | 4899 | 3600 | 47.76% | 4414 | 3600 | 37.54% | 31290 |
| 40_15_02 | 1983 | 4784 | 3600 | 49.08% | 4317 | 3600 | 39.29% | 29496 |
| 40_15_03 | 1635 | 4972 | 3600 | 58.76% | 4296 | 3600 | 36.66% | 30471 |
| 40_15_04 | 2071 | 4931 | 3600 | 48.52% | 4299 | 3600 | 38.24% | 29786 |
| 40_15_05 | 2163 | 4987 | 3600 | 47.56% | 4416 | 3600 | 38.68% | 30503 |
| 40_15_06 | 1996 | 4628 | 3600 | 48.59% | 4130 | 3600 | 37.63% | 29159 |
| 40_15_07 | 1706 | 4895 | 3600 | 55.13% | 4389 | 3600 | 39.53% | 29793 |
| 40_15_08 | 1849 | 4763 | 3600 | 51.60% | 4330 | 3600 | 38.11% | 30608 |
| 40_15_09 | 1812 | 4641 | 3600 | 52.79% | 4215 | 3600 | 40.17% | 29059 |
| 40_15_10 | 2094 | 4947 | 3600 | 48.35% | 4396 | 3600 | 38.88% | 29866 |
| **40_15_Average** | **1941.50** | **4844.70** | **3600** | **50.81%** | **4320.20** | **3600** | **38.47%** | **30003.10** |
| 50_15_01 | 2523 | 5922 | 3600 | 49.54% | 5061 | 3600 | 40.01% | 36632 |
| 50_15_02 | 2367 | 6290 | 3600 | 55.93% | 5184 | 3600 | 38.70% | 36862 |
| 50_15_03 | 2347 | 6221 | 3600 | 55.48% | 5220 | 3600 | 40.27% | 37090 |
| 50_15_04 | 2624 | 6313 | 3600 | 51.29% | 5310 | 3600 | 36.91% | 39575 |
| 50_15_05 | 2177 | 6020 | 3600 | 56.96% | 5197 | 3600 | 39.70% | 37372 |
| 50_15_06 | 2404 | 6283 | 3600 | 54.75% | 5284 | 3600 | 39.91% | 37845 |
| 50_15_07 | 2287 | 6160 | 3600 | 54.95% | 5338 | 3600 | 38.20% | 38921 |
| 50_15_08 | 2607 | 6158 | 3600 | 51.62% | 5251 | 3600 | 38.28% | 38477 |
| 50_15_09 | 2504 | 5930 | 3600 | 50.67% | 5236 | 3600 | 42.28% | 35680 |
| 50_15_10 | 2628 | 6374 | 3600 | 51.50% | 5202 | 3600 | 38.50% | 37789 |
| **50_15_Average** | **2446.80** | **6167.10** | **3600** | **53.27%** | **5228.30** | **3600** | **39.28%** | **37624.30** |
| 60_15_01 | 3152 | 7240 | 3600 | 51.68% | 6134 | 3600 | 38.54% | 45669 |
| 60_15_02 | 3066 | 7145 | 3600 | 51.84% | 6108 | 3600 | 39.96% | 44756 |
| 60_15_03 | 2808 | 7430 | 3600 | 55.85% | 6227 | 3600 | 41.03% | 45850 |
| 60_15_04 | 2837 | 7569 | 3600 | 57.99% | 6131 | 3600 | 42.29% | 43025 |
| 60_15_05 | 3170 | 7294 | 3600 | 50.49% | 6225 | 3600 | 41.40% | 45293 |
| 60_15_06 | 2955 | 7748 | 3600 | 55.31% | 6345 | 3600 | 41.75% | 46562 |
| 60_15_07 | 2898 | 7373 | 3600 | 54.94% | 6227 | 3600 | 42.28% | 44575 |
| 60_15_08 | 2905 | 7192 | 3600 | 54.25% | 6150 | 3600 | 42.13% | 43740 |
| 60_15_09 | 3015 | 7128 | 3600 | 51.34% | 5829 | 3600 | 38.05% | 44032 |
| 60_15_10 | 3045 | 7635 | 3600 | 55.40% | 6183 | 3600 | 39.84% | 45558 |
| **60_15_Average** | **2985.10** | **7375.40** | **3600** | **53.91%** | **6155.90** | **3600** | **40.73%** | **44906.00** |

## Table A.8. MILP-Prime and CP-Prime Comparison Table with Lower and Upper Bounds for 20 machines (VRF Instances)

| Instance | LB | MILP-Prime | Time (Seconds) | Gap % | CP-Prime | Time (Seconds) | Gap % | UB |
|---|---|---|---|---|---|---|---|---|
| 10_20_01 | 343 | 1913 | 1.812 | 0.00% | 1913 | 97.145 | 0.00% | 9382 |
| 10_20_02 | 523 | 1973 | 1.109 | 0.00% | 1973 | 55.923 | 0.00% | 9746 |
| 10_20_03 | 534 | 1989 | 2.109 | 0.00% | 1989 | 114.932 | 0.00% | 9475 |
| 10_20_04 | 363 | 1971 | 3.578 | 0.00% | 1971 | 78.640 | 0.00% | 9469 |
| 10_20_05 | 430 | 1979 | 1.938 | 0.00% | 1979 | 189.822 | 0.00% | 9411 |
| 10_20_06 | 627 | 2152 | 2.203 | 0.00% | 2152 | 78.785 | 0.00% | 10279 |
| 10_20_07 | 373 | 1893 | 1.094 | 0.00% | 1893 | 43.360 | 0.00% | 9790 |
| 10_20_08 | 373 | 1933 | 2.078 | 0.00% | 1933 | 131.791 | 0.00% | 9390 |
| 10_20_09 | 517 | 1941 | 1.641 | 0.00% | 1941 | 62.719 | 0.00% | 9943 |
| 10_20_10 | 460 | 1876 | 1.188 | 0.00% | 1876 | 60.680 | 0.00% | 9568 |
| **10_20_Average** | **454.30** | **1962.00** | **1.875** | **0.00%** | **1962.00** | **91.380** | **0.00%** | **9645.30** |
| 20_20_01 | 985 | 3128 | 3600 | 32.73% | 3082 | 3600 | 39.23% | 19780 |
| 20_20_02 | 1025 | 2894 | 3600 | 32.57% | 2964 | 3600 | 37.31% | 19573 |
| 20_20_03 | 1091 | 2935 | 3600 | 32.16% | 2935 | 3600 | 36.08% | 19864 |
| 20_20_04 | 1004 | 2863 | 3600 | 31.99% | 2852 | 3600 | 37.55% | 19380 |
| 20_20_05 | 1036 | 3116 | 3600 | 28.46% | 3078 | 3600 | 39.90% | 19893 |
| 20_20_06 | 957 | 3184 | 3600 | 37.84% | 3174 | 3600 | 39.79% | 19439 |
| 20_20_07 | 1078 | 2999 | 3600 | 25.24% | 2999 | 3600 | 37.75% | 20114 |
| 20_20_08 | 1057 | 2859 | 3600 | 36.47% | 2849 | 3600 | 33.20% | 19215 |
| 20_20_09 | 1142 | 3121 | 3600 | 35.75% | 3141 | 3600 | 35.88% | 20983 |
| 20_20_10 | 799 | 2885 | 3600 | 31.02% | 2884 | 3600 | 36.17% | 19302 |
| **20_20_Average** | **1017.40** | **2998.40** | **3600** | **32.42%** | **2995.80** | **3600** | **37.29%** | **19754.30** |
| 30_20_01 | 1424 | 4046 | 3600 | 50.00% | 3969 | 3600 | 43.86% | 28320 |
| 30_20_02 | 1600 | 4258 | 3600 | 48.30% | 4094 | 3600 | 40.99% | 29242 |
| 30_20_03 | 1363 | 4383 | 3600 | 53.52% | 4071 | 3600 | 40.24% | 30648 |
| 30_20_04 | 1354 | 4130 | 3600 | 53.17% | 3819 | 3600 | 39.28% | 28902 |
| 30_20_05 | 1482 | 4108 | 3600 | 50.85% | 3782 | 3600 | 38.95% | 28861 |
| 30_20_06 | 1491 | 4261 | 3600 | 48.39% | 3993 | 3600 | 40.65% | 29591 |
| 30_20_07 | 1368 | 4322 | 3600 | 53.03% | 4068 | 3600 | 40.86% | 28946 |
| 30_20_08 | 1451 | 4455 | 3600 | 52.39% | 4108 | 3600 | 39.68% | 29963 |
| 30_20_09 | 1548 | 4348 | 3600 | 49.12% | 4039 | 3600 | 39.74% | 30835 |
| 30_20_10 | 1493 | 4513 | 3600 | 51.16% | 4145 | 3600 | 41.64% | 29908 |
| **30_20_Average** | **1457.40** | **4282.40** | **3600** | **50.99%** | **4008.80** | **3600** | **40.59%** | **29521.60** |

**Table A.8.(Cont'd). MILP-Prime and CP-Prime Comparison Table with Lower and Upper Bounds for 20 machines (VRF Instances)**

| Instance | LB | MILP-Prime | Time (Seconds) | Gap % | CP-Prime | Time (Seconds) | Gap % | UB |
|---|---|---|---|---|---|---|---|---|
| 40_20_01 | 1991 | 5645 | 3600 | 56.91% | 5004 | 3600 | 38.83% | 39749 |
| 40_20_02 | 1933 | 5547 | 3600 | 55.07% | 4936 | 3600 | 43.03% | 39611 |
| 40_20_03 | 2033 | 5899 | 3600 | 54.70% | 5156 | 3600 | 45.67% | 39992 |
| 40_20_04 | 1636 | 5760 | 3600 | 60.43% | 4872 | 3600 | 40.93% | 40243 |
| 40_20_05 | 1933 | 5216 | 3600 | 54.67% | 4807 | 3600 | 43.48% | 38028 |
| 40_20_06 | 1964 | 5789 | 3600 | 55.43% | 5018 | 3600 | 44.42% | 38446 |
| 40_20_07 | 1859 | 5833 | 3600 | 58.32% | 5113 | 3600 | 44.32% | 39827 |
| 40_20_08 | 1928 | 5729 | 3600 | 54.66% | 5051 | 3600 | 42.84% | 39740 |
| 40_20_09 | 1996 | 5816 | 3600 | 53.76% | 5108 | 3600 | 42.56% | 40530 |
| 40_20_10 | 1855 | 5580 | 3600 | 55.64% | 4814 | 3600 | 41.88% | 38544 |
| **40_20_Average** | **1912.80** | **5681.40** | **3600** | **55.96%** | **4987.90** | **3600** | **42.80%** | **39471.00** |
| 50_20_01 | 2406 | 7359 | 3600 | 59.30% | 5914 | 3600 | 44.45% | 49263 |
| 50_20_02 | 2368 | 7229 | 3600 | 58.91% | 5982 | 3600 | 43.93% | 49291 |
| 50_20_03 | 2532 | 7164 | 3600 | 55.59% | 6036 | 3600 | 42.73% | 50327 |
| 50_20_04 | 2370 | 6950 | 3600 | 58.18% | 6092 | 3600 | 45.40% | 49819 |
| 50_20_05 | 2449 | 6994 | 3600 | 57.54% | 6115 | 3600 | 45.79% | 47908 |
| 50_20_06 | 2509 | 7465 | 3600 | 57.69% | 6140 | 3600 | 44.56% | 50932 |
| 50_20_07 | 2256 | 7076 | 3600 | 60.33% | 6043 | 3600 | 44.07% | 49551 |
| 50_20_08 | 2380 | 7090 | 3600 | 57.32% | 5998 | 3600 | 43.06% | 49620 |
| 50_20_09 | 2575 | 7281 | 3600 | 54.99% | 6180 | 3600 | 44.37% | 50936 |
| 50_20_10 | 2692 | 7181 | 3600 | 54.69% | 6091 | 3600 | 44.10% | 50468 |
| **50_20_Average** | **2453.70** | **7178.90** | **3600** | **57.45%** | **6059.10** | **3600** | **44.25%** | **49811.50** |
| 60_20_01 | 2926 | 8229 | 3600 | 57.13% | 7220 | 3600 | 46.36% | 59752 |
| 60_20_02 | 3101 | 8319 | 3600 | 54.87% | 7168 | 3600 | 45.70% | 60474 |
| 60_20_03 | 3236 | 8589 | 3600 | 55.65% | 7376 | 3600 | 45.95% | 61159 |
| 60_20_04 | 2914 | 8744 | 3600 | 61.12% | 7178 | 3600 | 44.71% | 59734 |
| 60_20_05 | 2998 | 8094 | 3600 | 55.97% | 6932 | 3600 | 44.60% | 59396 |
| 60_20_06 | 2904 | 8692 | 3600 | 59.42% | 6853 | 3600 | 43.09% | 59555 |
| 60_20_07 | 3024 | 8118 | 3600 | 55.42% | 7040 | 3600 | 43.59% | 60390 |
| 60_20_08 | 2922 | 8762 | 3600 | 59.95% | 7030 | 3600 | 44.96% | 59314 |
| 60_20_09 | 3090 | 8544 | 3600 | 56.89% | 7036 | 3600 | 44.37% | 59483 |
| 60_20_10 | 2701 | 8726 | 3600 | 62.74% | 6865 | 3600 | 43.51% | 59365 |
| **60_20_Average** | **2981.60** | **8481.70** | **3600** | **57.92%** | **7069.80** | **3600** | **44.68%** | **59862.20** |

## Table A.9. MILP and CP Comparison Table for 20 jobs (Taillard Instances)

| Instance | Opt. | MILP Result | Time (Seconds) | Gap % | CP Result | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|---|
| 20_5_01 | 1486 | 1497 | 3600 | 42.28% | 1486 | 3600 | 14.00% |
| 20_5_02 | 1528 | 1559 | 3600 | 50.67% | 1528 | 3600 | 11.13% |
| 20_5_03 | 1460 | 1485 | 3600 | 42.28% | 1462 | 3600 | 25.99% |
| 20_5_04 | 1588 | 1619 | 3600 | 44.71% | 1589 | 3600 | 18.19% |
| 20_5_05 | 1449 | 1465 | 3600 | 39.11% | 1449 | 3600 | 16.01% |
| 20_5_06 | 1481 | 1541 | 3600 | 43.34% | 1484 | 3600 | 19.20% |
| 20_5_07 | 1483 | 1513 | 3600 | 44.34% | 1485 | 3600 | 16.43% |
| 20_5_08 | 1482 | 1494 | 3600 | 38.95% | 1482 | 3600 | 18.62% |
| 20_5_09 | 1469 | 1498 | 3600 | 44.25% | 1469 | 3600 | 16.68% |
| 20_5_10 | 1377 | 1412 | 3600 | 44.75% | 1379 | 3600 | 19.94% |
| **20_5_Average** | **1480.30** | **1508.30** | **3600** | **43.47%** | **1481.30** | **3600** | **17.62%** |
| 20_10_01 | 2044 | 2069 | 3600 | 38.23% | 2055 | 3600 | 26.96% |
| 20_10_02 | 2166 | 2229 | 3600 | 41.18% | 2178 | 3600 | 27.46% |
| 20_10_03 | 1940 | 1963 | 3600 | 38.15% | 1951 | 3600 | 25.01% |
| 20_10_04 | 1811 | 1834 | 3600 | 38.05% | 1835 | 3600 | 27.08% |
| 20_10_05 | 1933 | 1949 | 3600 | 33.81% | 1933 | 3600 | 28.35% |
| 20_10_06 | 1892 | 1911 | 3600 | 39.03% | 1901 | 3600 | 28.67% |
| 20_10_07 | 1963 | 2029 | 3600 | 35.48% | 1963 | 3600 | 28.68% |
| 20_10_08 | 2057 | 2108 | 3600 | 41.46% | 2065 | 3600 | 30.61% |
| 20_10_09 | 1973 | 2021 | 3600 | 37.80% | 2017 | 3600 | 23.95% |
| 20_10_10 | 2051 | 2081 | 3600 | 37.53% | 2051 | 3600 | 27.25% |
| **20_10_Average** | **1983.00** | **2019.40** | **3600** | **38.07%** | **1994.90** | **3600** | **27.40%** |
| 20_20_01 | 2973 | 2984 | 3600 | 35.28% | 2976 | 3600 | 32.46% |
| 20_20_02 | 2852 | 2874 | 3600 | 33.57% | 2859 | 3600 | 36.24% |
| 20_20_03 | 3013 | 3068 | 3600 | 34.32% | 3044 | 3600 | 36.10% |
| 20_20_04 | 3001 | 3013 | 3600 | 34.11% | 3010 | 3600 | 35.78% |
| 20_20_05 | 3003 | 3003 | 3600 | 33.39% | 3032 | 3600 | 32.92% |
| 20_20_06 | 2998 | 3016 | 3600 | 31.26% | 2998 | 3600 | 34.39% |
| 20_20_07 | 3052 | 3082 | 3600 | 33.16% | 3078 | 3600 | 35.80% |
| 20_20_08 | 2839 | 2856 | 3600 | 31.96% | 2865 | 3600 | 32.43% |
| 20_20_09 | 3009 | 3049 | 3600 | 34.56% | 3101 | 3600 | 37.92% |
| 20_20_10 | 2979 | 3015 | 3600 | 33.88% | 2995 | 3600 | 34.76% |
| **20_20_Average** | **2971.90** | **2996.00** | **3600** | **33.55%** | **2995.80** | **3600** | **34.88%** |

## Table A.10. MILP and CP Comparison Table for 50 jobs (Taillard Instances)

| Instance | Opt. | MILP Result | Time (Seconds) | Gap % | CP Result | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|---|
| 50_5_01 | 3160 | 3758 | 3600 | 81.98% | 3212 | 3600 | 15.07% |
| 50_5_02 | 3432 | 3861 | 3600 | 82.02% | 3469 | 3600 | 18.25% |
| 50_5_03 | 3210 | 3719 | 3600 | 84.16% | 3251 | 3600 | 19.59% |
| 50_5_04 | 3338 | 3938 | 3600 | 83.74% | 3380 | 3600 | 18.11% |
| 50_5_05 | 3356 | 3762 | 3600 | 81.15% | 3408 | 3600 | 15.87% |
| 50_5_06 | 3346 | 3961 | 3600 | 81.84% | 3365 | 3600 | 15.84% |
| 50_5_07 | 3231 | 3749 | 3600 | 81.51% | 3256 | 3600 | 16.49% |
| 50_5_08 | 3235 | 3832 | 3600 | 83.22% | 3298 | 3600 | 18.28% |
| 50_5_09 | 3070 | 3722 | 3600 | 81.70% | 3107 | 3600 | 17.83% |
| 50_5_10 | 3317 | 3841 | 3600 | 83.24% | 3362 | 3600 | 17.19% |
| **50_5_Average** | **3269.50** | **3844.29** | **3600** | **82.89%** | **3310.80** | **3600** | **17.25%** |
| 50_10_01 | 4274 | 4923 | 3600 | 79.97% | 4353 | 3600 | 31.75% |
| 50_10_02 | 4177 | 4812 | 3600 | 79.77% | 4245 | 3600 | 33.22% |
| 50_10_03 | 4099 | 5025 | 3600 | 81.73% | 4182 | 3600 | 32.28% |
| 50_10_04 | 4399 | 5075 | 3600 | 77.10% | 4450 | 3600 | 31.17% |
| 50_10_05 | 4322 | 4999 | 3600 | 80.82% | 4374 | 3600 | 32.83% |
| 50_10_06 | 4289 | 5108 | 3600 | 82.57% | 4321 | 3600 | 30.87% |
| 50_10_07 | 4420 | 5279 | 3600 | 79.80% | 4454 | 3600 | 31.19% |
| 50_10_08 | 4318 | 5100 | 3600 | 78.70% | 4369 | 3600 | 31.17% |
| 50_10_09 | 4155 | 4985 | 3600 | 79.65% | 4232 | 3600 | 31.88% |
| 50_10_10 | 4283 | 5066 | 3600 | 79.49% | 4308 | 3600 | 29.29% |
| **50_10_Average** | **4273.60** | **5037.20** | **3600** | **79.96%** | **4328.80** | **3600** | **31.57%** |
| 50_20_01 | 6129 | 7192 | 3600 | 73.90% | 6255 | 3600 | 42.59% |
| 50_20_02 | 5725 | 6825 | 3600 | 74.56% | 5782 | 3600 | 38.86% |
| 50_20_03 | 5862 | 7065 | 3600 | 74.30% | 6029 | 3600 | 43.14% |
| 50_20_04 | 5788 | 6751 | 3600 | 75.97% | 5838 | 3600 | 40.85% |
| 50_20_05 | 5886 | 7129 | 3600 | 76.98% | 6018 | 3600 | 43.69% |
| 50_20_06 | 5863 | 6843 | 3600 | 74.13% | 6056 | 3600 | 41.55% |
| 50_20_07 | 5962 | 6827 | 3600 | 75.80% | 6094 | 3600 | 42.65% |
| 50_20_08 | 5926 | 7306 | 3600 | 76.93% | 6037 | 3600 | 42.97% |
| 50_20_09 | 5876 | 6839 | 3600 | 73.75% | 5939 | 3600 | 41.25% |
| 50_20_10 | 5957 | 7024 | 3600 | 76.90% | 6087 | 3600 | 41.88% |
| **50_20_Average** | **5897.40** | **6980.00** | **3600** | **75.32%** | **6013.50** | **3600** | **41.94%** |

## Table A.11. MILP and CP Comparison Table for 100 jobs (Taillard Instances)

| Instance | Opt. | MILP Result | Time (Seconds) | Gap % | CP Result | Time (Seconds) | Gap |
|---|---|---|---|---|---|---|---|
| 100_5_01 | 6361 | 8019 | 3600 | 92.61% | 6582 | 3600 | 1655% |
| 100_5_02 | 6212 | 8110 | 3600 | 92.14% | 6386 | 3600 | 17.82% |
| 100_5_03 | 6104 | 8003 | 3600 | 91.75% | 6250 | 3600 | 17.25% |
| 100_5_04 | 5999 | 7791 | 3600 | 91.61% | 6134 | 3600 | 18.50% |
| 100_5_05 | 6179 | 7945 | 3600 | 92.22% | 6351 | 3600 | 17.38% |
| 100_5_06 | 6056 | 8037 | 3600 | 92.09% | 6206 | 3600 | 17.29% |
| 100_5_07 | 6221 | 7769 | 3600 | 92.07% | 6399 | 3600 | 18.30% |
| 100_5_08 | 6109 | 7912 | 3600 | 92.46% | 6326 | 3600 | 19.65% |
| 100_5_09 | 6355 | 8370 | 3600 | 93.02% | 6514 | 3600 | 16.41% |
| 100_5_10 | 6365 | 8293 | 3600 | 93.60% | 6516 | 3600 | 18.49% |
| **100_5_Average** | **6196.10** | **8024.90** | **3600** | **92.47%** | **6366.40** | **3600** | **17.76%** |
| 100_10_01 | 8055 | no solution | 3600 | - | 8316 | 3600 | 30.53% |
| 100_10_02 | 7853 | no solution | 3600 | - | 8129 | 3600 | 34.11% |
| 100_10_03 | 8016 | no solution | 3600 | - | 8251 | 3600 | 31.40% |
| 100_10_04 | 8328 | no solution | 3600 | - | 8570 | 3600 | 32.54% |
| 100_10_05 | 7936 | no solution | 3600 | - | 8267 | 3600 | 33.82% |
| 100_10_06 | 7773 | no solution | 3600 | - | 8074 | 3600 | 34.37% |
| 100_10_07 | 7846 | no solution | 3600 | - | 8170 | 3600 | 31.85% |
| 100_10_08 | 7880 | no solution | 3600 | - | 8160 | 3600 | 31.54% |
| 100_10_09 | 8131 | no solution | 3600 | - | 8422 | 3600 | 30.34% |
| 100_10_10 | 8092 | no solution | 3600 | - | 8306 | 3600 | 29.73% |
| **100_10_Average** | **7991.00** | **no solution** | **3600** | **-** | **8266.50** | **3600** | **32.02%** |
| 100_20_01 | 10675 | no solution | 3600 | - | 11101 | 3600 | 46.53% |
| 100_20_02 | 10562 | no solution | 3600 | - | 11095 | 3600 | 44.82% |
| 100_20_03 | 10587 | no solution | 3600 | - | 10962 | 3600 | 43.79% |
| 100_20_04 | 10588 | no solution | 3600 | - | 10916 | 3600 | 43.54% |
| 100_20_05 | 10506 | no solution | 3600 | - | 11021 | 3600 | 44.10% |
| 100_20_06 | 10623 | no solution | 3600 | - | 11113 | 3600 | 44.18% |
| 100_20_07 | 10793 | no solution | 3600 | - | 11153 | 3600 | 45.42% |
| 100_20_08 | 10801 | no solution | 3600 | - | 11312 | 3600 | 45.28% |
| 100_20_09 | 10703 | no solution | 3600 | - | 11226 | 3600 | 45.98% |
| 100_20_10 | 10747 | no solution | 3600 | - | 11035 | 3600 | 42.17% |
| **100_20_Average** | **10658.50** | **no solution** | **3600** | **-** | **11093.40** | **3600,00** | **44.58%** |

## Table A.12. MILP and CP Comparison Table for 200 jobs (Taillard Instances)

| Instance | Opt. | MILP Result | Time (Seconds) | Gap % | CP Result | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|---|
| 200_10_01 | 15225 | no solution | 3600 | - | 16144 | 3600 | 32.72% |
| 200_10_02 | 14990 | no solution | 3600 | - | 15963 | 3600 | 34.50% |
| 200_10_03 | 15257 | no solution | 3600 | - | 16261 | 3600 | 32.85% |
| 200_10_04 | 15103 | no solution | 3600 | - | 16171 | 3600 | 32.93% |
| 200_10_05 | 15088 | no solution | 3600 | - | 16078 | 3600 | 34.73% |
| 200_10_06 | 14976 | no solution | 3600 | - | 15883 | 3600 | 34.99% |
| 200_10_07 | 15277 | no solution | 3600 | - | 16300 | 3600 | 33.42% |
| 200_10_08 | 15133 | no solution | 3600 | - | 16208 | 3600 | 33.89% |
| 200_10_09 | 14985 | no solution | 3600 | - | 15956 | 3600 | 34.64% |
| 200_10_10 | 15213 | no solution | 3600 | - | 16239 | 3600 | 34.32% |
| **200_10_Average** | **15124.70** | **no solution** | **3600** | **-** | **16120.30** | **3600** | **34.03%** |
| 200_20_01 | 19531 | no solution | 3600 | - | 21051 | 3600 | 47.51% |
| 200_20_02 | 19942 | no solution | 3600 | - | 21565 | 3600 | 48.95% |
| 200_20_03 | 19759 | no solution | 3600 | - | 21255 | 3600 | 47.17% |
| 200_20_04 | 19759 | no solution | 3600 | - | 21227 | 3600 | 47.34% |
| 200_20_05 | 19697 | no solution | 3600 | - | 21278 | 3600 | 47.45% |
| 200_20_06 | 19826 | no solution | 3600 | - | 21589 | 3600 | 48.31% |
| 200_20_07 | 19946 | no solution | 3600 | - | 21824 | 3600 | 48.33% |
| 200_20_08 | 19872 | no solution | 3600 | - | 21488 | 3600 | 47.80% |
| 200_20_09 | 19784 | no solution | 3600 | - | 21358 | 3600 | 48.24% |
| 200_20_10 | 19768 | no solution | 3600 | - | 21351 | 3600 | 47.32% |
| **200_20_Average** | **19788.40** | **no solution** | **3600** | **-** | **21398.60** | **3600** | **47.84%** |

# APPENDIX B – Computational Results for $(F_m|nwt|\sum C_{iM})$

## Table B.1. MILP and CP Comparison Table for 5 machines (VRF Instances)

| Instance | MILP Result | Time (Seconds) | Gap % | CP Result | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|
| 10_5_01 | 4117 | 1.516 | 0.00% | 4117 | 12.703 | 0.00% |
| 10_5_02 | 4045 | 0.688 | 0.00% | 4045 | 5.232 | 0.00% |
| 10_5_03 | 4380 | 0.875 | 0.00% | 4380 | 10.211 | 0.00% |
| 10_5_04 | 4199 | 0.641 | 0.00% | 4199 | 6.189 | 0.00% |
| 10_5_05 | 4666 | 0.875 | 0.00% | 4666 | 6.886 | 0.00% |
| 10_5_06 | 5524 | 2.484 | 0.00% | 5524 | 23.437 | 0.00% |
| 10_5_07 | 4798 | 1.719 | 0.00% | 4798 | 15.215 | 0.00% |
| 10_5_08 | 4080 | 1.000 | 0.00% | 4080 | 8.290 | 0.00% |
| 10_5_09 | 5118 | 1.118 | 0.00% | 5118 | 7.215 | 0.00% |
| 10_5_10 | 4136 | 1.703 | 0.00% | 4136 | 13.828 | 0.00% |
| **10_5_Average** | **4506.30** | **1.262** | **0.00%** | **4506.30** | **10.921** | **0.00%** |
| 20_5_01 | 15038 | 3600 | 45.64% | 15020 | 3600 | 27.05% |
| 20_5_02 | 16310 | 3600 | 48.72% | 16054 | 3600 | 24.41% |
| 20_5_03 | 17314 | 3600 | 48.54% | 17314 | 3600 | 24.97% |
| 20_5_04 | 14078 | 3600 | 43.94% | 14078 | 3600 | 23.63% |
| 20_5_05 | 17139 | 3600 | 48.63% | 17098 | 3600 | 19.28% |
| 20_5_06 | 14576 | 3600 | 46.54% | 14567 | 3600 | 31.43% |
| 20_5_07 | 14309 | 3600 | 42.28% | 14294 | 3600 | 27.17% |
| 20_5_08 | 14249 | 3600 | 43.51% | 14249 | 3600 | 23.65% |
| 20_5_09 | 16479 | 3600 | 48.77% | 16197 | 3600 | 17.04% |
| 20_5_10 | 17063 | 3600 | 49.86% | 17043 | 3600 | 27.95% |
| **20_5_Average** | **15655.50** | **3600** | **46.64%** | **15591.40** | **3600** | **24.66%** |
| 30_5_01 | 34786 | 3600 | 69.07% | 34095 | 3600 | 31.81% |
| 30_5_02 | 31540 | 3600 | 69.33% | 28816 | 3600 | 30.42% |
| 30_5_03 | 30827 | 3600 | 66.33% | 30479 | 3600 | 28.65% |
| 30_5_04 | 34950 | 3600 | 69.98% | 32834 | 3600 | 23.64% |
| 30_5_05 | 32664 | 3600 | 66.18% | 31415 | 3600 | 29.92% |
| 30_5_06 | 33738 | 3600 | 68.40% | 33211 | 3600 | 32.76% |
| 30_5_07 | 31528 | 3600 | 68.21% | 30651 | 3600 | 21.46% |
| 30_5_08 | 33068 | 3600 | 67.51% | 32466 | 3600 | 28.17% |
| 30_5_09 | 34061 | 3600 | 68.86% | 31822 | 3600 | 32.02% |
| 30_5_10 | 33146 | 3600 | 68.73% | 32280 | 3600 | 35.78% |
| **30_5_Average** | **33030.80** | **3600** | **68.26%** | **31806.90** | **3600** | **29.46%** |

## Table B.1.(Cont'd.) MILP and CP Comparison Table for 5 machines (VRF Instances)

| Instance | MILP Result | Time (Seconds) | Gap % | CP Result | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|
| 40_5_01 | 57083 | 3600 | 75.84% | 54385 | 3600 | 23.88% |
| 40_5_02 | 62153 | 3600 | 73.68% | 57918 | 3600 | 27.43% |
| 40_5_03 | 53951 | 3600 | 75.83% | 51229 | 3600 | 27.77% |
| 40_5_04 | 58685 | 3600 | 77.75% | 53447 | 3600 | 31.70% |
| 40_5_05 | 61597 | 3600 | 77.37% | 55761 | 3600 | 30.76% |
| 40_5_06 | 58087 | 3600 | 77.70% | 51668 | 3600 | 27.80% |
| 40_5_07 | 57573 | 3600 | 76.85% | 52269 | 3600 | 27.82% |
| 40_5_08 | 65180 | 3600 | 78.05% | 58177 | 3600 | 28.24% |
| 40_5_09 | 61246 | 3600 | 77.41% | 55669 | 3600 | 29.72% |
| 40_5_10 | 58935 | 3600 | 76.01% | 54916 | 3600 | 25.40% |
| **40_5_Average** | **59449.00** | **3600** | **76.65%** | **54543.90** | **3600** | **28.05%** |
| 50_5_01 | 94488 | 3600 | 79.19% | 87966 | 3600 | 26.16% |
| 50_5_02 | 92657 | 3600 | 82.22% | 82880 | 3600 | 29.00% |
| 50_5_03 | 93318 | 3600 | 82.66% | 84893 | 3600 | 36.12% |
| 50_5_04 | 96914 | 3600 | 77.66% | 82391 | 3600 | 27.66% |
| 50_5_05 | 94661 | 3600 | 82.29% | 84389 | 3600 | 30.32% |
| 50_5_06 | 96009 | 3600 | 83.07% | 82772 | 3600 | 26.70% |
| 50_5_07 | 86259 | 3600 | 82.26% | 76369 | 3600 | 30.90% |
| 50_5_08 | 84286 | 3600 | 80.89% | 80534 | 3600 | 33.48% |
| 50_5_09 | 84316 | 3600 | 79.46% | 77903 | 3600 | 33.10% |
| 50_5_10 | 91841 | 3600 | 81.36% | 83864 | 3600 | 28.49% |
| **50_5_Average** | **91474.90** | **3600** | **81.11%** | **82396.10** | **3600** | **30.19%** |
| 60_5_01 | 131383 | 3600 | 84.72% | 116575 | 3600 | 2989% |
| 60_5_02 | 121169 | 3600 | 83.71% | 112053 | 3600 | 37.19% |
| 60_5_03 | 126360 | 3600 | 81.52% | 113745 | 3600 | 34.18% |
| 60_5_04 | 132960 | 3600 | 83.13% | 116630 | 3600 | 35.00% |
| 60_5_05 | 133941 | 3600 | 85.52% | 115036 | 3600 | 33.39% |
| 60_5_06 | 127392 | 3600 | 85.20% | 110802 | 3600 | 36.01% |
| 60_5_07 | 144766 | 3600 | 86.17% | 116420 | 3600 | 29.83% |
| 60_5_08 | 142678 | 3600 | 85.63% | 122113 | 3600 | 28.73% |
| 60_5_09 | 126528 | 3600 | 85.10% | 111726 | 3600 | 32.45% |
| 60_5_10 | 131600 | 3600 | 84.65% | 116138 | 3600 | 17.99% |
| **60_5_Average** | **131877.70** | **3600** | **84.54%** | **115123.80** | **3600** | **31.47%** |

## Table B.2. MILP and CP Comparison Table for 10 machines (VRF Instances)

| Instance | MILP Result | Time (Seconds) | Gap % | CP Result | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|
| 10_10_01 | 7994 | 1.109 | 0.00% | 7994 | 25.365 | 0.00% |
| 10_10_02 | 7936 | 1.235 | 0.00% | 7936 | 40.060 | 0.00% |
| 10_10_03 | 7719 | 0.703 | 0.00% | 7719 | 22.434 | 0.00% |
| 10_10_04 | 7236 | 0.687 | 0.00% | 7236 | 18.481 | 0.00% |
| 10_10_05 | 8394 | 1.609 | 0.00% | 8394 | 51.397 | 0.00% |
| 10_10_06 | 7553 | 1.266 | 0.00% | 7553 | 39.634 | 0.00% |
| 10_10_07 | 7441 | 0.625 | 0.00% | 7441 | 21.928 | 0.00% |
| 10_10_08 | 7776 | 0.656 | 0.00% | 7776 | 23.520 | 0.00% |
| 10_10_09 | 7840 | 0.750 | 0.00% | 7840 | 32.240 | 0.00% |
| 10_10_10 | 8015 | 1.156 | 0.00% | 8015 | 36.536 | 0.00% |
| **10_10_Average** | **7790.40** | **0.980** | **0.00%** | **7790.40** | **31.160** | **0.00%** |
| 20_10_01 | 23788 | 3600 | 44.06% | 23565 | 3600 | 27.11% |
| 20_10_02 | 23293 | 3600 | 40.20% | 23091 | 3600 | 25.21% |
| 20_10_03 | 23897 | 3600 | 37.72% | 23751 | 3600 | 29.20% |
| 20_10_04 | 23006 | 3600 | 41.29% | 22529 | 3600 | 31.25% |
| 20_10_05 | 24367 | 3600 | 40.73% | 24367 | 3600 | 26.86% |
| 20_10_06 | 25687 | 3600 | 44.36% | 24928 | 3600 | 32.53% |
| 20_10_07 | 24132 | 3600 | 40.49% | 24132 | 3600 | 30.03% |
| 20_10_08 | 23078 | 3600 | 39.80% | 23015 | 3600 | 24.41% |
| 20_10_09 | 23662 | 3600 | 40.33% | 23363 | 3600 | 31.39% |
| 20_10_10 | 23333 | 3600 | 40.02% | 23337 | 3600 | 31.01% |
| **20_10_Average** | **23824.30** | **3600** | **40.90%** | **23607.80** | **3600** | **28.90%** |
| 30_10_01 | 48415 | 3600 | 62.54% | 44934 | 3600 | 34.03% |
| 30_10_02 | 50114 | 3600 | 61.06% | 47390 | 3600 | 32.47% |
| 30_10_03 | 49065 | 3600 | 60.84% | 47725 | 3600 | 32.93% |
| 30_10_04 | 48345 | 3600 | 62.11% | 45973 | 3600 | 38.65% |
| 30_10_05 | 49899 | 3600 | 61.34% | 48373 | 3600 | 36.04% |
| 30_10_06 | 49160 | 3600 | 59.88% | 47229 | 3600 | 34.43% |
| 30_10_07 | 45753 | 3600 | 54.57% | 45569 | 3600 | 34.51% |
| 30_10_08 | 44529 | 3600 | 60.63% | 43038 | 3600 | 35.39% |
| 30_10_09 | 46231 | 3600 | 60.94% | 44671 | 3600 | 37.12% |
| 30_10_10 | 44229 | 3600 | 59.21% | 43173 | 3600 | 38.63% |
| **30_10_Average** | **47574.00** | **3600** | **60.31%** | **45807.50** | **3600** | **35.42%** |

**Table B.2.(Cont'd.) MILP and CP Comparison Table for 10 machines (VRF Instances)**

| Instance | MILP Result | Time (Seconds) | Gap % | CP Result | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|
| 40_10_01 | 78846 | 3600 | 69.48% | 76417 | 3600 | 37.61% |
| 40_10_02 | 83339 | 3600 | 7144% | 74534 | 3600 | 37.33% |
| 40_10_03 | 80989 | 3600 | 71.31% | 74206 | 3600 | 37.49% |
| 40_10_04 | 86420 | 3600 | 71.51% | 78784 | 3600 | 40.86% |
| 40_10_05 | 88325 | 3600 | 67.22% | 76434 | 3600 | 39.59% |
| 40_10_06 | 84022 | 3600 | 70.96% | 79540 | 3600 | 37.03% |
| 40_10_07 | 84013 | 3600 | 71.29% | 77928 | 3600 | 38.02% |
| 40_10_08 | 79387 | 3600 | 67.15% | 74595 | 3600 | 38.56% |
| 40_10_09 | 79686 | 3600 | 69.90% | 74597 | 3600 | 40.47% |
| 40_10_10 | 81645 | 3600 | 70.49% | 75962 | 3600 | 39.19% |
| **40_10_Average** | **82667.20** | **3600** | **70.08%** | **76299.70** | **3600** | **38.62%** |
| 50_10_01 | 120036 | 3600 | 75.57% | 112242 | 3600 | 41.48% |
| 50_10_02 | 127181 | 3600 | 76.69% | 114210 | 3600 | 38.71% |
| 50_10_03 | 124736 | 3600 | 75.72% | 113219 | 3600 | 39.39% |
| 50_10_04 | 127514 | 3600 | 76.21% | 119059 | 3600 | 41.13% |
| 50_10_05 | 124697 | 3600 | 64.88% | 118704 | 3600 | 34.92% |
| 50_10_06 | 122750 | 3600 | 68.58% | 109833 | 3600 | 32.20% |
| 50_10_07 | 123951 | 3600 | 74.87% | 110347 | 3600 | 41.10% |
| 50_10_08 | 130818 | 3600 | 76.37% | 118392 | 3600 | 41.18% |
| 50_10_09 | 133077 | 3600 | 77.42% | 117628 | 3600 | 43.26% |
| 50_10_10 | 131778 | 3600 | 69.65% | 115446 | 3600 | 37.64% |
| **50_10_Average** | **126653.80** | **3600** | **73.60%** | **114908.00** | **3600** | **39.10%** |
| 60_10_01 | 180093 | 3600 | 80.14% | 159559 | 3600 | 44.66% |
| 60_10_02 | 188791 | 3600 | 80.37% | 162564 | 3600 | 37.28% |
| 60_10_03 | 187931 | 3600 | 81.08% | 156946 | 3600 | 41.77% |
| 60_10_04 | 185298 | 3600 | 80.92% | 161533 | 3600 | 44.41% |
| 60_10_05 | 188368 | 3600 | 80.22% | 165545 | 3600 | 44.27% |
| 60_10_06 | 189026 | 3600 | 80.57% | 166416 | 3600 | 44.19% |
| 60_10_07 | 182751 | 3600 | 79.38% | 163330 | 3600 | 39.86% |
| 60_10_08 | 174641 | 3600 | 79.76% | 156487 | 3600 | 42.10% |
| 60_10_09 | 178934 | 3600 | 80.10% | 159501 | 3600 | 38.84% |
| 60_10_10 | 182960 | 3600 | 80.23% | 159669 | 3600 | 43.78% |
| **60_10_Average** | **183879.30** | **3600** | **80.28%** | **161155.00** | **3600** | **42.12%** |

## Table B.3. MILP and CP Comparison Table for 15 machines (VRF Instances)

| Instance | MILP Result | Time (Seconds) | Gap % | CP Result | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|
| 10_15_01 | 10089 | 0.922 | 0.00% | 10089 | 58.739 | 0.00% |
| 10_15_02 | 11178 | 1.219 | 0.00% | 11178 | 75.550 | 0.00% |
| 10_15_03 | 11272 | 1.312 | 0.00% | 11272 | 100.877 | 0.00% |
| 10_15_04 | 10799 | 1.031 | 0.00% | 10799 | 60.795 | 0.00% |
| 10_15_05 | 11110 | 1.250 | 0.00% | 11110 | 70.873 | 0.00% |
| 10_15_06 | 10431 | 0.859 | 0.00% | 10431 | 58.791 | 0.00% |
| 10_15_07 | 10972 | 1.484 | 0.00% | 10972 | 79.354 | 0.00% |
| 10_15_08 | 11125 | 1.219 | 0.00% | 11125 | 60.905 | 0.00% |
| 10_15_09 | 11121 | 0.953 | 0.00% | 11121 | 65.490 | 0.00% |
| 10_15_10 | 11613 | 1.422 | 0.00% | 11613 | 98.221 | 0.00% |
| **10_15_Average** | **10971.00** | **1.167** | **0.00%** | **10971.00** | **72.960** | **0.00%** |
| 20_15_01 | 32430 | 3600 | 37.16% | 32430 | 3600 | 32.18% |
| 20_15_02 | 32469 | 3600 | 38.31% | 32195 | 3600 | 30.51% |
| 20_15_03 | 31293 | 3600 | 38.92% | 31301 | 3600 | 34.46% |
| 20_15_04 | 30802 | 3600 | 35.96% | 30904 | 3600 | 35.20% |
| 20_15_05 | 30979 | 3600 | 35.23% | 30619 | 3600 | 26.92% |
| 20_15_06 | 32184 | 3600 | 36.45% | 31998 | 3600 | 32.51% |
| 20_15_07 | 31856 | 3600 | 36.54% | 31527 | 3600 | 30.62% |
| 20_15_08 | 31085 | 3600 | 36.38% | 30877 | 3600 | 34.20% |
| 20_15_09 | 29978 | 3600 | 32.08% | 29978 | 3600 | 24.01% |
| 20_15_10 | 31951 | 3600 | 37.25% | 31617 | 3600 | 32.82% |
| **20_15_Average** | **31502.70** | **3600** | **36.43%** | **31344.60** | **3600** | **31.34%** |
| 30_15_01 | 65120 | 3600 | 59.04% | 60543 | 3600 | 38.33% |
| 30_15_02 | 62996 | 3600 | 58.10% | 56992 | 3600 | 35.77% |
| 30_15_03 | 61114 | 3600 | 57.16% | 57640 | 3600 | 38.56% |
| 30_15_04 | 66561 | 3600 | 58.53% | 60522 | 3600 | 35.71% |
| 30_15_05 | 61364 | 3600 | 55.73% | 59937 | 3600 | 36.19% |
| 30_15_06 | 63221 | 3600 | 57.58% | 61263 | 3600 | 40.84% |
| 30_15_07 | 64671 | 3600 | 58.45% | 60288 | 3600 | 36.78% |
| 30_15_08 | 65066 | 3600 | 57.63% | 62295 | 3600 | 40.95% |
| 30_15_09 | 60183 | 3600 | 57.14% | 58185 | 3600 | 35.81% |
| 30_15_10 | 63033 | 3600 | 53.79% | 60234 | 3600 | 38.71% |
| **30_15_Average** | **63332.90** | **3600** | **57.32%** | **59789.90** | **3600** | **37.77%** |

**Table B.3.(Cont'd.) MILP and CP Comparison Table for 15 machines (VRF Instances)**

| Instance | MILP Result | Time (Seconds) | Gap % | CP Result | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|
| 40_15_01 | 110688 | 3600 | 65.39% | 103869 | 3600 | 41.78% |
| 40_15_02 | 101090 | 3600 | 65.36% | 97620 | 3600 | 41.70% |
| 40_15_03 | 109249 | 3600 | 66.99% | 99282 | 3600 | 40.81% |
| 40_15_04 | 104618 | 3600 | 66.58% | 98218 | 3600 | 42.89% |
| 40_15_05 | 106457 | 3600 | 66.27% | 100130 | 3600 | 42.71% |
| 40_15_06 | 98429 | 3600 | 65.18% | 93959 | 3600 | 41.44% |
| 40_15_07 | 106620 | 3600 | 66.89% | 99241 | 3600 | 42.06% |
| 40_15_08 | 103602 | 3600 | 64.97% | 97891 | 3600 | 41.12% |
| 40_15_09 | 96189 | 3600 | 64.20% | 92764 | 3600 | 40.29% |
| 40_15_10 | 105002 | 3600 | 66.53% | 98336 | 3600 | 39.65% |
| **40_15_Average** | **104194.40** | **3600** | **65.84%** | **98131.00** | **3600** | **41.45%** |
| 50_15_01 | 147187 | 3600 | 71.31% | 136074 | 3600 | 44.14% |
| 50_15_02 | 163146 | 3600 | 73.63% | 146125 | 3600 | 42.33% |
| 50_15_03 | 157116 | 3600 | 72.50% | 145073 | 3600 | 45.15% |
| 50_15_04 | 167239 | 3600 | 72.44% | 149217 | 3600 | 41.48% |
| 50_15_05 | 158126 | 3600 | 72.52% | 140097 | 3600 | 43.46% |
| 50_15_06 | 153500 | 3600 | 71.30% | 146440 | 3600 | 44.70% |
| 50_15_07 | 164343 | 3600 | 72.48% | 149659 | 3600 | 42.66% |
| 50_15_08 | 162446 | 3600 | 72.33% | 147661 | 3600 | 44.11% |
| 50_15_09 | 160378 | 3600 | 73.81% | 143908 | 3600 | 45.94% |
| 50_15_10 | 154426 | 3600 | 71.96% | 140941 | 3600 | 43.99% |
| **50_15_Average** | **158790.70** | **3600** | **72.43%** | **144519.50** | **3600** | **43.80%** |
| 60_15_01 | 229822 | 3600 | 76.92% | 200477 | 3600 | 43.13% |
| 60_15_02 | 221518 | 3600 | 76.72% | 199486 | 3600 | 45.22% |
| 60_15_03 | 231894 | 3600 | 77.37% | 200729 | 3600 | 46.21% |
| 60_15_04 | 216310 | 3600 | 75.50% | 193267 | 3600 | 45.29% |
| 60_15_05 | 230155 | 3600 | 77.53% | 198195 | 3600 | 46.60% |
| 60_15_06 | 235768 | 3600 | 77.28% | 209383 | 3600 | 48.74% |
| 60_15_07 | 227100 | 3600 | 77.12% | 200456 | 3600 | 46.51% |
| 60_15_08 | 216370 | 3600 | 74.67% | 193986 | 3600 | 47.12% |
| 60_15_09 | 216465 | 3600 | 76.41% | 197097 | 3600 | 46.89% |
| 60_15_10 | 227747 | 3600 | 76.79% | 199175 | 3600 | 44.48% |
| **60_15_Average** | **225314.90** | **3600** | **76.63%** | **199225.10** | **3600** | **46.02%** |

### Table B.4. MILP and CP Comparison Table for 20 machines (VRF Instances)

| Instance | MILP Result | Time (Seconds) | Gap % | CP Result | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|
| 10_20_01 | 13418 | 0.953 | 0.00% | 13418 | 112.112 | 0.00% |
| 10_20_02 | 14098 | 0.953 | 0.00% | 14098 | 150.869 | 0.00% |
| 10_20_03 | 13523 | 0.922 | 0.00% | 13523 | 130.233 | 0.00% |
| 10_20_04 | 14150 | 1.687 | 0.00% | 14150 | 193.526 | 0.00% |
| 10_20_05 | 13766 | 1.437 | 0.00% | 13766 | 177.039 | 0.00% |
| 10_20_06 | 15574 | 1.062 | 0.00% | 15574 | 193.254 | 0.00% |
| 10_20_07 | 14474 | 1.875 | 0.00% | 14474 | 205.521 | 0.00% |
| 10_20_08 | 14046 | 2.031 | 0.00% | 14046 | 197.313 | 0.00% |
| 10_20_09 | 14417 | 1.391 | 0.00% | 14417 | 173.098 | 0.00% |
| 10_20_10 | 13728 | 1.297 | 0.00% | 13728 | 152.416 | 0.00% |
| **10_20_Average** | **14119.40** | **1.361** | **0.00%** | **14119.40** | **168.538** | **0.00%** |
| 20_20_01 | 38550 | 3600 | 33.60% | 38789 | 3600 | 33.59% |
| 20_20_02 | 36529 | 3600 | 31.63% | 36529 | 3600 | 30.28% |
| 20_20_03 | 37628 | 3600 | 31.79% | 38009 | 3600 | 29.88% |
| 20_20_04 | 38924 | 3600 | 35.66% | 37348 | 3600 | 32.45% |
| 20_20_05 | 39422 | 3600 | 32.76% | 39279 | 3600 | 32.97% |
| 20_20_06 | 39631 | 3600 | 35.70% | 39641 | 3600 | 33.46% |
| 20_20_07 | 37854 | 3600 | 30.46% | 37702 | 3600 | 29.18% |
| 20_20_08 | 36759 | 3600 | 32.40% | 36646 | 3600 | 27.59% |
| 20_20_09 | 41404 | 3600 | 35.71% | 40347 | 3600 | 31.33% |
| 20_20_10 | 38098 | 3600 | 34.17% | 37207 | 3600 | 27.08% |
| **20_20_Average** | **38479.90** | **3600** | **33.39%** | **38149.70** | **3600** | **30.78%** |
| 30_20_01 | 71779 | 3600 | 52.84% | 68325 | 3600 | 38.22% |
| 30_20_02 | 73917 | 3600 | 53.37% | 71282 | 3600 | 35.36% |
| 30_20_03 | 78216 | 3600 | 53.54% | 74964 | 3600 | 39.67% |
| 30_20_04 | 73022 | 3600 | 53.40% | 68823 | 3600 | 33.25% |
| 30_20_05 | 72818 | 3600 | 52.61% | 69348 | 3600 | 35.38% |
| 30_20_06 | 74333 | 3600 | 52.82% | 72178 | 3600 | 38.22% |
| 30_20_07 | 76332 | 3600 | 55.03% | 72027 | 3600 | 38.37% |
| 30_20_08 | 76960 | 3600 | 54.08% | 73393 | 3600 | 37.41% |
| 30_20_09 | 78937 | 3600 | 54.10% | 75099 | 3600 | 38.86% |
| 30_20_10 | 75328 | 3600 | 52.80% | 73585 | 3600 | 36.22% |
| **30_20_Average** | **75164,20** | **3600** | **53.46%** | **71902.40** | **3600** | **37.10%** |

**Table B.4.(Cont'd.) MILP and CP Comparison Table for 20 machines (VRF Instances)**

| Instance | MILP Result | Time (Seconds) | Gap % | CP Result | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|
| 40_20_01 | 119352 | 3600 | 61.65% | 111832 | 3600 | 34.89% |
| 40_20_02 | 118070 | 3600 | 61.34% | 112999 | 3600 | 40.51% |
| 40_20_03 | 126812 | 3600 | 63.44% | 117738 | 3600 | 43.36% |
| 40_20_04 | 122914 | 3600 | 62.48% | 114795 | 3600 | 41.65% |
| 40_20_05 | 114661 | 3600 | 61.78% | 108700 | 3600 | 42.31% |
| 40_20_06 | 118647 | 3600 | 62.63% | 113060 | 3600 | 42.25% |
| 40_20_07 | 129603 | 3600 | 64.16% | 118708 | 3600 | 41.74% |
| 40_20_08 | 125133 | 3600 | 63.43% | 115644 | 3600 | 41.47% |
| 40_20_09 | 124553 | 3600 | 61.62% | 120029 | 3600 | 42.82% |
| 40_20_10 | 122453 | 3600 | 63.64% | 112120 | 3600 | 41.35% |
| **40_20_Average** | **122219.80** | **3600** | **62.62%** | **114562.50** | **3600** | **41.24%** |
| 50_20_01 | 181084 | 3600 | 68.98% | 168138 | 3600 | 45.68% |
| 50_20_02 | 186786 | 3600 | 69.81% | 168182 | 3600 | 45.94% |
| 50_20_03 | 189326 | 3600 | 66.46% | 172722 | 3600 | 46.01% |
| 50_20_04 | 192111 | 3600 | 69.96% | 168541 | 3600 | 45.09% |
| 50_20_05 | 177003 | 3600 | 68.80% | 164533 | 3600 | 46.39% |
| 50_20_06 | 187298 | 3600 | 69.08% | 170986 | 3600 | 44.83% |
| 50_20_07 | 185242 | 3600 | 69.15% | 173809 | 3600 | 47.81% |
| 50_20_08 | 188369 | 3600 | 70.09% | 164240 | 3600 | 43.96% |
| 50_20_09 | 187632 | 3600 | 68.93% | 174257 | 3600 | 45.59% |
| 50_20_10 | 182296 | 3600 | 68.27% | 167737 | 3600 | 44.33% |
| **50_20_Average** | **185714.70** | **3600** | **68.95%** | **169314.50** | **3600** | **45.56%** |
| 60_20_01 | 262874 | 3600 | 73.98% | 234288 | 3600 | 48.25% |
| 60_20_02 | 252294 | 3600 | 72.65% | 231177 | 3600 | 46.85% |
| 60_20_03 | 262336 | 3600 | 73.41% | 240352 | 3600 | 47.67% |
| 60_20_04 | 264779 | 3600 | 74.14% | 238675 | 3600 | 47.13% |
| 60_20_05 | 245218 | 3600 | 72.39% | 225762 | 3600 | 46.63% |
| 60_20_06 | 259639 | 3600 | 74.08% | 229040 | 3600 | 47.40% |
| 60_20_07 | 256533 | 3600 | 73.28% | 239125 | 3600 | 47.02% |
| 60_20_08 | 261945 | 3600 | 74.23% | 232828 | 3600 | 48.01% |
| 60_20_09 | 258469 | 3600 | 73.84% | 229811 | 3600 | 47.59% |
| 60_20_10 | 257208 | 3600 | 73.59% | 228803 | 3600 | 46.78% |
| **60_20_Average** | **258129.50** | **3600** | **73.56%** | **232986.10** | **3600** | **47.33%** |

### Table B.5. MILP and CP Comparison Table for 20 jobs (Taillard Instances)

| Instance | MILP Result | Time (Seconds) | Gap % | CP Result | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|
| 20_5_01 | 15698 | 3600 | 46.27% | 15674 | 3600 | 19.19% |
| 20_5_02 | 17503 | 3600 | 52.01% | 17270 | 3600 | 23.72% |
| 20_5_03 | 16035 | 3600 | 49.26% | 15821 | 3600 | 30.86% |
| 20_5_04 | 17976 | 3600 | 48.52% | 17970 | 3600 | 26.24% |
| 20_5_05 | 15331 | 3600 | 46.61% | 15317 | 3600 | 17.41% |
| 20_5_06 | 15501 | 3600 | 45.39% | 15501 | 3600 | 27.21% |
| 20_5_07 | 15712 | 3600 | 48.41% | 15706 | 3600 | 28.96% |
| 20_5_08 | 15959 | 3600 | 47.92% | 16023 | 3600 | 23.32% |
| 20_5_09 | 16634 | 3600 | 48.61% | 16385 | 3600 | 27.53% |
| 20_5_10 | 15347 | 3600 | 49.60% | 15371 | 3600 | 29.03% |
| **20_5_Average** | **16169.60** | **3600** | **48.26%** | **16103.80** | **3600** | **25.35%** |
| 20_10_01 | 25397 | 3600 | 42.92% | 25205 | 3600 | 32.07% |
| 20_10_02 | 26663 | 3600 | 42.29% | 26371 | 3600 | 29.58% |
| 20_10_03 | 22910 | 3600 | 39.10% | 22910 | 3600 | 27.49% |
| 20_10_04 | 22762 | 3600 | 44.49% | 22243 | 3600 | 33.05% |
| 20_10_05 | 23482 | 3600 | 41.51% | 23269 | 3600 | 35.76% |
| 20_10_06 | 22199 | 3600 | 41.38% | 22011 | 3600 | 31.17% |
| 20_10_07 | 22210 | 3600 | 40.20% | 21939 | 3600 | 30.02% |
| 20_10_08 | 24427 | 3600 | 42.35% | 24205 | 3600 | 32.49% |
| 20_10_09 | 23967 | 3600 | 40.78% | 23501 | 3600 | 28.56% |
| 20_10_10 | 24721 | 3600 | 38.94% | 24715 | 3600 | 27.80% |
| **20_10_Average** | **23873.80** | **3600** | **41.40%** | **23636.90** | **3600** | **30.80%** |
| 20_20_01 | 39142 | 3600 | 34.17% | 38728 | 3600 | 25.89% |
| 20_20_02 | 37643 | 3600 | 34.42% | 37571 | 3600 | 34.17% |
| 20_20_03 | 38574 | 3600 | 33.38% | 38382 | 3600 | 29.40% |
| 20_20_04 | 39341 | 3600 | 35.60% | 38802 | 3600 | 31.18% |
| 20_20_05 | 39167 | 3600 | 33.22% | 39012 | 3600 | 25.93% |
| 20_20_06 | 39182 | 3600 | 35.10% | 38618 | 3600 | 32.51% |
| 20_20_07 | 39855 | 3600 | 36.17% | 39663 | 3600 | 31.57% |
| 20_20_08 | 37027 | 3600 | 32.82% | 37000 | 3600 | 28.27% |
| 20_20_09 | 39267 | 3600 | 33.91% | 39228 | 3600 | 32.93% |
| 20_20_10 | 37977 | 3600 | 35.00% | 37931 | 3600 | 30.12% |
| **20_20_Average** | **38717.50** | **3600** | **34.38%** | **38493.50** | **3600** | **30.20%** |

## Table B.6. MILP and CP Comparison Table for 50 jobs (Taillard Instances)

| Instance | MILP Result | Time (Seconds) | Gap % | CP Result | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|
| 50_5_01 | 87996 | 3600 | 81.93% | 76530 | 3600 | 28.99% |
| 50_5_02 | 95874 | 3600 | 82.11% | 85619 | 3600 | 32.47% |
| 50_5_03 | 87715 | 3600 | 81.77% | 78440 | 3600 | 33.53% |
| 50_5_04 | 93070 | 3600 | 82.02% | 83296 | 3600 | 32.23% |
| 50_5_05 | 99298 | 3600 | 83.16% | 84746 | 3600 | 31.55% |
| 50_5_06 | 93416 | 3600 | 81.91% | 81127 | 3600 | 29.05% |
| 50_5_07 | 91346 | 3600 | 82.25% | 79299 | 3600 | 27.19% |
| 50_5_08 | 90865 | 3600 | 82.07% | 79772 | 3600 | 31.47% |
| 50_5_09 | 81988 | 3600 | 79.46% | 76122 | 3600 | 32.34% |
| 50_5_10 | 93507 | 3600 | 81.93% | 85986 | 3600 | 32.80% |
| **50_5_Average** | **91507.50** | **3600** | **81.86%** | **81093.70** | **3600** | **31.16%** |
| 50_10_01 | 129431 | 3600 | 76.72% | 115198 | 3600 | 43.19% |
| 50_10_02 | 125020 | 3600 | 76.66% | 113511 | 3600 | 42.33% |
| 50_10_03 | 125711 | 3600 | 76.80% | 107260 | 3600 | 41.09% |
| 50_10_04 | 125382 | 3600 | 71.36% | 116509 | 3600 | 41.78% |
| 50_10_05 | 132574 | 3600 | 76.72% | 117023 | 3600 | 43.34% |
| 50_10_06 | 125256 | 3600 | 75.99% | 114474 | 3600 | 37.71% |
| 50_10_07 | 124452 | 3600 | 75.31% | 119119 | 3600 | 37.18% |
| 50_10_08 | 127132 | 3600 | 73.00% | 116537 | 3600 | 40.64% |
| 50_10_09 | 124187 | 3600 | 76.07% | 111266 | 3600 | 40.09% |
| 50_10_10 | 122804 | 3600 | 74.69% | 114622 | 3600 | 36.05% |
| **50_10_Average** | **126194.90** | **3600** | **75.33%** | **114551.90** | **3600** | **40.34%** |
| 50_20_01 | 186314 | 3600 | 68.14% | 174557 | 3600 | 44.95% |
| 50_20_02 | 182986 | 3600 | 69.36% | 162979 | 3600 | 42.50% |
| 50_20_03 | 178097 | 3600 | 68.77% | 164065 | 3600 | 46.60% |
| 50_20_04 | 186633 | 3600 | 69.41% | 165958 | 3600 | 43.26% |
| 50_20_05 | 184516 | 3600 | 69.37% | 169117 | 3600 | 47.42% |
| 50_20_06 | 179826 | 3600 | 68.15% | 163859 | 3600 | 43.43% |
| 50_20_07 | 182040 | 3600 | 68.85% | 168343 | 3600 | 44.82% |
| 50_20_08 | 189041 | 3600 | 70.26% | 171248 | 3600 | 46.23% |
| 50_20_09 | 179280 | 3600 | 65.16% | 169472 | 3600 | 42.84% |
| 50_20_10 | 182768 | 3600 | 68.58% | 173979 | 3600 | 44.42% |
| **50_20_Average** | **183150.10** | **3600** | **68.61%** | **168357.70** | **3600** | **44.65%** |

## Table B.7. MILP and CP Comparison Table for 100 jobs (Taillard Instances)

| Instance | MILP Result | Time (Seconds) | Gap % | CP Result | Time (Seconds) | Gap |
|---|---|---|---|---|---|---|
| 100_5_01 | 385065 | 3600 | 91.57% | 313165 | 3600 | 32.19% |
| 100_5_02 | 352042 | 3600 | 87.89% | 304434 | 3600 | 34.38% |
| 100_5_03 | 383576 | 3600 | 91.96% | 300643 | 3600 | 35.07% |
| 100_5_04 | 314768 | 3600 | 86.06% | 283011 | 3600 | 33.96% |
| 100_5_05 | 356990 | 3600 | 90.32% | 302073 | 3600 | 33.60% |
| 100_5_06 | 353622 | 3600 | 88.66% | 294999 | 3600 | 37.32% |
| 100_5_07 | 387178 | 3600 | 91.95% | 306007 | 3600 | 36.56% |
| 100_5_08 | 377355 | 3600 | 91.94% | 295676 | 3600 | 39.08% |
| 100_5_09 | 389823 | 3600 | 91.69% | 312497 | 3600 | 35.91% |
| 100_5_10 | 390391 | 3600 | 91.71% | 310077 | 3600 | 35.59% |
| **100_5_Average** | **369081.00** | **3600** | **90.38%** | **302258.20** | **3600** | **35.37%** |
| 100_10_01 | no solution | 3600 | - | 426032 | 3600 | 43.89% |
| 100_10_02 | no solution | 3600 | - | 408264 | 3600 | 49.07% |
| 100_10_03 | no solution | 3600 | - | 419751 | 3600 | 45.31% |
| 100_10_04 | no solution | 3600 | - | 431381 | 3600 | 46.01% |
| 100_10_05 | no solution | 3600 | - | 409270 | 3600 | 46.63% |
| 100_10_06 | no solution | 3600 | - | 400684 | 3600 | 48.47% |
| 100_10_07 | no solution | 3600 | - | 401456 | 3600 | 43.54% |
| 100_10_08 | no solution | 3600 | - | 413451 | 3600 | 43.95% |
| 100_10_09 | no solution | 3600 | - | 431039 | 3600 | 42.24% |
| 100_10_10 | no solution | 3600 | - | 428113 | 3600 | 45.98% |
| **100_10_Average** | **no solution** | **3600** | **-** | **416944.10** | **3600** | **45.51%** |
| 100_20_01 | no solution | 3600 | - | 593720 | 3600 | 53.57% |
| 100_20_02 | no solution | 3600 | - | 590592 | 3600 | 53.44% |
| 100_20_03 | no solution | 3600 | - | 586146 | 3600 | 50.36% |
| 100_20_04 | no solution | 3600 | - | 597415 | 3600 | 51.59% |
| 100_20_05 | no solution | 3600 | - | 580374 | 3600 | 49.66% |
| 100_20_06 | no solution | 3600 | - | 591962 | 3600 | 53.67% |
| 100_20_07 | no solution | 3600 | - | 611832 | 3600 | 55.49% |
| 100_20_08 | no solution | 3600 | - | 601038 | 3600 | 51.43% |
| 100_20_09 | no solution | 3600 | - | 589340 | 3600 | 53.36% |
| 100_20_10 | no solution | 3600 | - | 596446 | 3600 | 48.15% |
| **100_20_Average** | **no solution** | **3600** | **-** | **593886.50** | **3600** | **52.07%** |

## Table B.8. MILP and CP Comparison Table for 200 jobs (Taillard Instances)

| Instance | MILP Result | Time (Seconds) | Gap % | CP Result | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|
| 200_10_01 | no solution | 3600 | - | 1614812 | 3600 | 47.95% |
| 200_10_02 | no solution | 3600 | - | 1667646 | 3600 | 53.13% |
| 200_10_03 | no solution | 3600 | - | 1653710 | 3600 | 48.77% |
| 200_10_04 | no solution | 3600 | - | 1591963 | 3600 | 50.35% |
| 200_10_05 | no solution | 3600 | - | 1634213 | 3600 | 51.31% |
| 200_10_06 | no solution | 3600 | - | 1577967 | 3600 | 51.55% |
| 200_10_07 | no solution | 3600 | - | 1633800 | 3600 | 51.83% |
| 200_10_08 | no solution | 3600 | - | 1631609 | 3600 | 50.42% |
| 200_10_09 | no solution | 3600 | - | 1601298 | 3600 | 51.53% |
| 200_10_10 | no solution | 3600 | - | 1616365 | 3600 | 50.41% |
| **200_10_Average** | **no solution** | **3600** | **-** | **1622338.30** | **3600** | **50.73%** |
| 200_20_01 | no solution | 3600 | - | 2197865 | 3600 | 5913% |
| 200_20_02 | no solution | 3600 | - | 2237183 | 3600 | 59.58% |
| 200_20_03 | no solution | 3600 | - | 2205949 | 3600 | 58.32% |
| 200_20_04 | no solution | 3600 | - | 2270101 | 3600 | 60.24% |
| 200_20_05 | no solution | 3600 | - | 2260614 | 3600 | 60.15% |
| 200_20_06 | no solution | 3600 | - | 2243069 | 3600 | 59.96% |
| 200_20_07 | no solution | 3600 | - | 2225255 | 3600 | 58.08% |
| 200_20_08 | no solution | 3600 | - | 2237216 | 3600 | 59.33% |
| 200_20_09 | no solution | 3600 | - | 2187614 | 3600 | 58.79% |
| 200_20_10 | no solution | 3600 | - | 2211536 | 3600 | 58.88% |
| **200_20_Average** | **no solution** | **3600** | **-** | **2227640.20** | **3600** | **59.25%** |

# APPENDIX C – Computational Results for $(F_m|nwt|\sum T_i)$

## Table C.1. MILP and CP Comparison Table for 20 jobs (Taillard Instances)

| Instance | MILP Result | Time (Seconds) | Gap % | CP Result | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|
| 20_5_01 | 4005 | 3600 | 90.98% | 3874 | 3600 | 88.36% |
| 20_5_02 | 4167 | 3600 | 76.69% | 4167 | 3600 | 93.52% |
| 20_5_03 | 5830 | 3600 | 79.18% | 5648 | 3600 | 93.66% |
| 20_5_04 | 4660 | 3600 | 76.25% | 4408 | 3600 | 94.94% |
| 20_5_05 | 3292 | 3600 | 70.04% | 3292 | 3600 | 92.77% |
| 20_5_06 | 5523 | 3600 | 78.55% | 5395 | 3600 | 92.96% |
| 20_5_07 | 3469 | 3600 | 82.69% | 3365 | 3600 | 94.00% |
| 20_5_08 | 5237 | 3600 | 78.44% | 5224 | 3600 | 81.34% |
| 20_5_09 | 4056 | 3600 | 85.92% | 3974 | 3600 | 92.63% |
| 20_5_10 | 4886 | 3600 | 86.19% | 4795 | 3600 | 91.89% |
| **20_5_Average** | **4512.50** | **3600** | **80.49%** | **4414.20** | **3600** | **91.61%** |
| 20_10_01 | 757 | 3600 | 94.84% | 716 | 3600 | 100.00% |
| 20_10_02 | 2269 | 3600 | 47,59% | 2258 | 3600 | 99.16% |
| 20_10_03 | 1551 | 3600 | 62.66% | 1518 | 3600 | 100.00% |
| 20_10_04 | 1338 | 1795 | 0.00% | 1338 | 3600 | 98.73% |
| 20_10_05 | 1359 | 3600 | 81.89% | 1270 | 3600 | 100.00% |
| 20_10_06 | 2760 | 3600 | 39.38% | 2940 | 3600 | 90.99% |
| 20_10_07 | 1752 | 3600 | 87.83% | 1752 | 3600 | 100.00% |
| 20_10_08 | 124 | 2892 | 0.00% | 124 | 3600 | 100.00% |
| 20_10_09 | 2996 | 3600 | 43.89% | 2891 | 3600 | 97.82% |
| 20_10_10 | 3145 | 3600 | 31.09% | 3145 | 3600 | 95.64% |
| **20_10_Average** | **1805.10** | **3348.70** | **48.92%** | **1795.20** | **3600** | **98.23%** |
| 20_20_01 | 0 | 3.484 | 0.00% | 0 | 8.700 | 0.00% |
| 20_20_02 | 216 | 2.078 | 0.00% | 216 | 90.659 | 0.00% |
| 20_20_03 | 0 | 2.969 | 0.00% | 0 | 7.531 | 0.00% |
| 20_20_04 | 0 | 10.515 | 0.00% | 0 | 35.382 | 0.00% |
| 20_20_05 | 3934 | 400.719 | 0.00% | 4022 | 3600.000 | 95.57% |
| 20_20_06 | 2541 | 28.578 | 0.00% | 2541 | 887.407 | 0.00% |
| 20_20_07 | 364 | 57.047 | 0.00% | 364 | 572.882 | 0.00% |
| 20_20_08 | 844 | 2.328 | 0.00% | 844 | 158.326 | 0.00% |
| 20_20_09 | 258 | 259.579 | 0.00% | 258 | 574.038 | 0.00% |
| 20_20_10 | 0 | 1.110 | 0.00% | 0 | 0.614 | 0.00% |
| **20_20_Average** | **815.70** | **76.840** | **0.00%** | **824.50** | **593.55** | **9.56%** |

## Table C.2. MILP and CP Comparison Table for 50 jobs (Taillard Instances)

| Instance | MILP Result | Time (Seconds) | Gap % | CP Result | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|
| 50_5_01 | 58311 | 3600 | 9894% | 49587 | 3600 | 47.78% |
| 50_5_02 | 64752 | 3600 | 99.36% | 51522 | 3600 | 53.43% |
| 50_5_03 | 55886 | 3600 | 98.96% | 49998 | 3600 | 58.80% |
| 50_5_04 | 60802 | 3600 | 98.59% | 55199 | 3600 | 52.37% |
| 50_5_05 | 63260 | 3600 | 99.22% | 55109 | 3600 | 51.15% |
| 50_5_06 | 55011 | 3600 | 99.18% | 48688 | 3600 | 51.01% |
| 50_5_07 | 56771 | 3600 | 99.36% | 50723 | 3600 | 44.97% |
| 50_5_08 | 56477 | 3600 | 99.20% | 48953 | 3600 | 53.05% |
| 50_5_09 | 57092 | 3600 | 99.66% | 47075 | 3600 | 56.07% |
| 50_5_10 | 61747 | 3600 | 98.64% | 55906 | 3600 | 50.52% |
| **50_5_Average** | **59010.90** | **3600** | **99.11%** | **51276.00** | **3600** | **51.92%** |
| 50_10_01 | 62282 | 3600 | 9945% | 54132 | 3600 | 98.30% |
| 50_10_02 | 63309 | 3600 | 99.26% | 55523 | 3600 | 91.50% |
| 50_10_03 | 62432 | 3600 | 99.28% | 52531 | 3600 | 89.71% |
| 50_10_04 | 63571 | 3600 | 99.57% | 50409 | 3600 | 97.80% |
| 50_10_05 | 66647 | 3600 | 98.79% | 57375 | 3600 | 95.63% |
| 50_10_06 | 64629 | 3600 | 99.72% | 53513 | 3600 | 84.29% |
| 50_10_07 | 71458 | 3600 | 99.10% | 58342 | 3600 | 78.22% |
| 50_10_08 | 66788 | 3600 | 99.60% | 58678 | 3600 | 84.16% |
| 50_10_09 | 73544 | 3600 | 98.64% | 60616 | 3600 | 81.36% |
| 50_10_10 | 65992 | 3600 | 99.02% | 54591 | 3600 | 83.61% |
| **50_10_Average** | **66065.20** | **3600** | **99.24%** | **55571.00** | **3600** | **88.46%** |
| 50_20_01 | 67509 | 3600 | 9914% | 57719 | 3600 | 99.44% |
| 50_20_02 | 57540 | 3600 | 99.13% | 51268 | 3600 | 99.73% |
| 50_20_03 | 72294 | 3600 | 99.40% | 58871 | 3600 | 99.60% |
| 50_20_04 | 71022 | 3600 | 9833% | 53738 | 3600 | 99.36% |
| 50_20_05 | 77549 | 3600 | 98.71% | 59081 | 3600 | 99.69% |
| 50_20_06 | 67632 | 3600 | 99.64% | 49353 | 3600 | 99.86% |
| 50_20_07 | 70883 | 3600 | 99.26% | 59037 | 3600 | 99.63% |
| 50_20_08 | 71641 | 3600 | 99.69% | 53200 | 3600 | 99.89% |
| 50_20_09 | 65027 | 3600 | 99.16% | 54899 | 3600 | 99.85% |
| 50_20_10 | 77607 | 3600 | 99.24% | 69084 | 3600 | 99.10% |
| **50_20_Average** | **69870.40** | **3600** | **99.17%** | **56625.00** | **3600** | **99.62%** |

**Table C.3. MILP and CP Comparison Table for 100 jobs (Taillard Instances)**

| Instance | MILP Result | Time (Seconds) | Gap % | CP Result | Time (Seconds) | Gap |
|---|---|---|---|---|---|---|
| 100_5_01 | 298416 | 3600 | 99.82% | 244835 | 3600 | 40.97% |
| 100_5_02 | 302432 | 3600 | 99.08% | 244738 | 3600 | 44.66% |
| 100_5_03 | 300131 | 3600 | 99.66% | 240511 | 3600 | 43.67% |
| 100_5_04 | 276616 | 3600 | 99.58% | 231855 | 3600 | 43.68% |
| 100_5_05 | 302965 | 3600 | 99.72% | 237423 | 3600 | 42.98% |
| 100_5_06 | 310924 | 3600 | 99.85% | 231990 | 3600 | 47.44% |
| 100_5_07 | 296315 | 3600 | 99.72% | 249601 | 3600 | 46.80% |
| 100_5_08 | 306647 | 3600 | 99.73% | 239252 | 3600 | 49.56% |
| 100_5_09 | 304255 | 3600 | 99.80% | 245070 | 3600 | 46.02% |
| 100_5_10 | 309836 | 3600 | 99.73% | 243843 | 3600 | 43.54% |
| **100_5_Average** | **300853.70** | **3600** | **99.67%** | **240911.80** | **3600** | **44.93%** |
| 100_10_01 | no solution | 3600 | - | 298584 | 3600 | 64.93% |
| 100_10_02 | no solution | 3600 | - | 285989 | 3600 | 69.76% |
| 100_10_03 | no solution | 3600 | - | 292524 | 3600 | 63.22% |
| 100_10_04 | no solution | 3600 | - | 308846 | 3600 | 67.05% |
| 100_10_05 | no solution | 3600 | - | 290366 | 3600 | 67.88% |
| 100_10_06 | no solution | 3600 | - | 283485 | 3600 | 70.34% |
| 100_10_07 | no solution | 3600 | - | 289199 | 3600 | 64.75% |
| 100_10_08 | no solution | 3600 | - | 301357 | 3600 | 65.07% |
| 100_10_09 | no solution | 3600 | - | 296423 | 3600 | 58.96% |
| 100_10_10 | no solution | 3600 | - | 296844 | 3600 | 67.89% |
| **100_10_Average** | **no solution** | **3600** | **-** | **294361.70** | **3600** | **65.99%** |
| 100_20_01 | no solution | 3600 | - | 356508 | 3600 | 95.07% |
| 100_20_02 | no solution | 3600 | - | 346425 | 3600 | 98.51% |
| 100_20_03 | no solution | 3600 | - | 355974 | 3600 | 87.04% |
| 100_20_04 | no solution | 3600 | - | 367858 | 3600 | 87.35% |
| 100_20_05 | no solution | 3600 | - | 362186 | 3600 | 83.02% |
| 100_20_06 | no solution | 3600 | - | 348878 | 3600 | 95.01% |
| 100_20_07 | no solution | 3600 | - | 354352 | 3600 | 95.94% |
| 100_20_08 | no solution | 3600 | - | 369639 | 3600 | 92.69% |
| 100_20_09 | no solution | 3600 | - | 359996 | 3600 | 89.98% |
| 100_20_10 | no solution | 3600 | - | 356731 | 3600 | 83.40% |
| **100_20_Average** | **no solution** | **3600** | **-** | **357854.70** | **3600** | **90.80%** |

## Table C.4. MILP and CP Comparison Table for 200 jobs (Taillard Instances)

| Instance | MILP Result | Time (Seconds) | Gap % | CP Result | Time (Seconds) | Gap % |
|---|---|---|---|---|---|---|
| 200_10_01 | no solution | 3600 | - | 1381304 | 3600 | 57.80% |
| 200_10_02 | no solution | 3600 | - | 1360948 | 3600 | 61.10% |
| 200_10_03 | no solution | 3600 | - | 1401695 | 3600 | 57.62% |
| 200_10_04 | no solution | 3600 | - | 1370115 | 3600 | 60.33% |
| 200_10_05 | no solution | 3600 | - | 1368269 | 3600 | 59.93% |
| 200_10_06 | no solution | 3600 | - | 1406425 | 3600 | 63.65% |
| 200_10_07 | no solution | 3600 | - | 1430629 | 3600 | 62.35% |
| 200_10_08 | no solution | 3600 | - | 1398493 | 3600 | 60.67% |
| 200_10_09 | no solution | 3600 | - | 1386090 | 3600 | 61.78% |
| 200_10_10 | no solution | 3600 | - | 1374473 | 3600 | 59.93% |
| **200_10_Average** | **no solution** | **3600** | **-** | **1387844.10** | **3600** | **60.52%** |
| 200_20_01 | no solution | 3600 | - | 1767739 | 3600 | 75.70% |
| 200_20_02 | no solution | 3600 | - | 1806047 | 3600 | 76.63% |
| 200_20_03 | no solution | 3600 | - | 1721282 | 3600 | 76.34% |
| 200_20_04 | no solution | 3600 | - | 1783066 | 3600 | 77.73% |
| 200_20_05 | no solution | 3600 | - | 1705869 | 3600 | 77.32% |
| 200_20_06 | no solution | 3600 | - | 1825765 | 3600 | 77.76% |
| 200_20_07 | no solution | 3600 | - | 1837149 | 3600 | 76.13% |
| 200_20_08 | no solution | 3600 | - | 1762582 | 3600 | 76.84% |
| 200_20_09 | no solution | 3600 | - | 1815825 | 3600 | 76.82% |
| 200_20_10 | no solution | 3600 | - | 1777477 | 3600 | 76.95% |
| **200_20_Average** | **no solution** | **3600** | **-** | **1780280.10** | **3600** | **76.82%** |

# APPENDIX D –Small Sized Instances for $(F_m|nwt|\sum T_i, TEC)$

## Table D.1. Truncated Instances with a sets of 5x5, 5x10, and 5x20

```
jobs = 5;
machines = 5;

//Taillard_5_5_1
PTime = [[77,56,89,78,53] [36,70,45,91,35] [91,61,1,9,72] [77,14,47,40,87]
[94,77,40,31,28]];
DueDate = [314,326,336,179,341];

//Taillard_5_5_2
PTime = [[88,10,49,83,35] [23,54,36,92,77] [43,92,87,48,78]
[43,91,11,13,80] [50,37,5,98,72]];
DueDate = [302,289,223,329,391];

//Taillard_5_5_3
PTime = [[79,58,46,10,33] [25,79,44,43,32] [38,17,1,75,7] [22,8,76,70,30]
[27,26,59,84,75]];
DueDate = [88,55,75,113,0];

//Taillard_5_5_4
PTime = [[53,93,90,65,64] [39,62,54,73,90] [79,77,67,21,63]
[29,14,98,51,67] [48,25,20,44,18]];
DueDate = [153,305,212,264,297];

//Taillard_5_5_5
PTime = [[86,92,93,47,48] [46,2,95,57,62] [78,85,74,62,10] [72,14,4,90,99]
[34,48,97,37,62]];
DueDate = [347,296,353,114,400];

//Taillard_5_5_6
PTime = [[11,27,89,58,20] [18,33,75,59,69] [42,57,60,85,45]
[41,23,37,51,85] [75,99,65,97,8]];
DueDate = [74,208,44,78,105];

//Taillard_5_5_7
PTime = [[9,1,81,90,54] [27,77,98,3,39] [42,52,12,99,33] [11,28,84,73,86]
[50,65,11,87,37]];
DueDate = [113,291,336,232,278];

//Taillard_5_5_8
PTime = [[34,5,86,28,8] [20,48,35,39,91] [47,43,93,21,55] [74,87,40,59,59]
[62,84,6,18,89]];
DueDate = [35,113,102,89,5];

//Taillard_5_5_9
PTime = [[37,59,65,70,94] [36,16,94,3,98] [64,15,57,30,97] [98,69,8,1,61]
[89,9,13,46,37]];
DueDate = [217,272,228,248,279];

//Taillard_5_5_10
PTime = [[27,79,22,93,38] [41,51,34,97,93] [20,40,77,91,40]
[39,32,47,32,49] [91,16,39,26,90]];
DueDate = [188,175,155,36,165];
```

## Table D.1.(Cont'd.) Truncated Instances with a sets of 5x5, 5x10, and 5x20

```
jobs = 5;
machines = 10;

//Taillard_5_10_1
PTime = [[21,3,52,88,66,11,8,18,15,84] [21,34,7,76,70,57,27,95,56,95]
[83,87,98,47,84,77,2,18,70,91] [94,43,36,78,58,86,13,5,64,91]
[6,79,85,90,5,56,11,4,14,3]];
DueDate = [747,890,663,981,353];

//Taillard_5_10_2
PTime = [[80,59,59,31,30,53,93,90,65,64] [13,83,70,64,88,19,79,92,97,38]
[77,85,10,9,22,62,77,13,25,46] [43,71,66,1,39,72,48,38,96,69]
[14,59,70,73,11,57,98,15,56,81]];
DueDate = [618,899,945,781,705];

//Taillard_5_10_3
PTime = [[15,59,15,46,60,47,41,38,34,22] [18,7,26,17,87,32,9,26,33,34]
[37,40,53,89,59,80,42,37,85,30] [93,54,13,55,15,31,63,38,61,90]
[64,83,17,3,94,38,10,62,70,17]];
DueDate = [483,289,718,608,449];

//Taillard_5_10_4
PTime = [[94,3,39,1,63,86,44,19,55,67] [6,43,28,83,50,19,85,12,68,66]
[31,52,77,38,4,40,50,29,88,13] [31,87,21,89,61,22,13,2,36,27]
[32,21,26,29,51,57,74,22,46,50]];
DueDate = [506,656,548,703,408];

//Taillard_5_10_5
PTime=[[13,34,52,84,66,2,40,20,7,54] [17,12,32,87,90,93,29,61,6,31]
[26,16,87,99,15,92,57,93,39,37] [39,73,13,14,5,77,65,31,58,59]
[5,93,2,18,90,73,21,81,89,32]];
DueDate = [799,554,745,434,891];

//Taillard_5_10_6
PTime = [[77,46,79,22,20,96,75,1,37,14] [77,85,18,72,67,44,56,1,90,14]
[11,67,2,2,40,56,77,47,60,64] [36,39,46,58,36,46,14,23,65,30]
[92,25,12,46,60,83,3,21,12,33]];
DueDate = [488,587,484,393,520];

//Taillard_5_10_7
PTime = [[64,43,9,38,2,79,16,85,89,69] [95,46,20,21,20,12,25,28,77,43]
[65,66,7,15,81,56,8,51,55,81] [31,45,82,58,27,9,82,9,30,98]
[84,49,49,36,52,6,5,94,89,92]];
DueDate = [779,830,699,471,618];

//Taillard_5_10_8
PTime = [[9,91,96,73,37,28,32,27,4,83] [71,13,80,53,9,21,34,97,68,14]
[12,27,17,10,89,49,47,57,28,67] [85,88,54,97,93,60,73,1,6,31]
[33,5,83,84,95,52,17,18,67,69]];
DueDate = [627,666,403,736,541];

//Taillard_5_10_9Cropped
PTime = [[37,4,43,28,17,18,99,97,21,29] [37,92,18,94,47,47,34,10,98,20]
[24,26,66,10,84,74,28,51,74,29] [74,80,60,91,16,65,50,98,70,98]
[36,24,26,38,48,91,58,33,95,68]];
```

**Table D.1.(Cont'd.) Truncated Instances with a sets of 5x5, 5x10, and 5x20**

```
DueDate = [700,441,513,759,545];

//Taillard_5_10_10
PTime = [[26,92,20,61,91,58,70,20,86,36] [90,34,86,84,90,91,50,19,88,67]
[63,80,97,56,82,81,64,74,26,84] [37,71,12,38,84,31,99,87,33,80]
[30,75,32,47,5,74,11,52,61,60]];
DueDate = [560,630,721,730,648];

jobs = 5;
machines = 20;

//Taillard_5_20_1
PTime = [[81,73,48,99,8,41,51,82,25,25,55,58,16,16,48,69,94,62,7,55]
[48,38,70,21,15,33,92,98,73,95,79,55,59,94,88,1,65,38,10,8]
[43,65,87,80,93,36,89,61,26,3,85,22,2,67,41,66,7,50,4,74]
[1,93,85,4,39,80,46,28,73,2,64,83,17,3,94,38,10,62,70,17]
[87,1,72,19,88,74,88,22,18,41,35,44,41,71,71,72,38,97,49,19]];
DueDate = [1013,1599,1343,1269,1188];

//Taillard_5_20_2
PTime = [[45,83,86,3,15,8,73,6,55,8,22,44,17,1,77,23,42,79,30,22]
[51,62,19,3,11,77,58,64,74,30,72,54,29,75,78,64,95,40,86,8]
[31,52,77,38,4,40,50,29,88,13,46,3,17,48,21,20,26,25,6,25]
[36,1,81,66,7,82,55,77,67,29,12,23,25,60,15,92,26,78,10,83]
[5,72,77,42,94,52,98,13,47,86,1,70,46,67,61,94,86,64,29,87]];
DueDate = [739,1151,1176,1508,1433];

//Taillard_5_20_3
PTime = [[52,2,2,2,99,1,87,28,91,29,16,91,3,28,62,87,3,11,74,30]
[79,85,44,16,37,58,88,88,11,2,42,38,58,78,25,38,94,7,26,92]
[44,19,85,81,22,58,25,3,36,77,94,66,44,91,73,23,4,85,11,3]
[85,12,32,85,67,64,90,41,57,15,72,86,24,6,16,97,82,87,72,41]
[13,42,90,94,36,11,9,51,43,87,97,59,39,35,62,71,92,82,24,38]];
DueDate = [1441,1411,1250,1241,1075];

//Taillard_5_20_4
PTime = [[25,53,50,32,95,64,16,66,55,62,1,24,6,27,60,51,88,63,97,70]
[55,86,49,56,94,85,38,85,49,90,54,87,33,87,40,5,40,50,7,49]
[70,77,19,8,58,92,91,79,81,65,86,10,33,87,38,32,40,68,18,27]
[3,17,5,95,26,36,72,34,32,19,39,73,13,14,5,77,65,31,58,59]
[82,91,98,91,5,72,64,29,52,6,18,68,9,17,28,47,24,5,50,34]];
DueDate = [1008,1167,1300,773,1357];

//Taillard_5_20_5
PTime = [[40,94,46,90,69,69,3,18,98,12,25,20,34,43,2,47,6,56,69,85]
[86,28,89,63,61,7,79,27,98,97,50,72,23,13,60,44,17,13,41,14]
[29,7,51,26,99,90,96,46,99,54,16,10,97,71,70,52,4,74,20,76]
[36,46,18,48,76,31,24,58,55,95,82,42,25,22,35,3,10,27,70,58]
[61,46,75,20,61,22,5,80,22,86,43,19,98,72,14,70,94,46,61,25]];
DueDate = [1063,1166,1098,1061,1020];

//Taillard_5_20_6
PTime = [[66,21,45,56,49,39,13,34,22,53,40,17,72,50,99,50,26,99,61,1]
[20,63,48,24,87,13,69,25,22,8,25,7,69,7,62,59,46,79,37,91]
[1,16,71,71,45,49,83,18,14,92,10,19,18,37,10,7,82,50,43,20]
```

**Table D.1.(Cont'd.) Truncated Instances with a sets of 5x5, 5x10, and 5x20**

```
[99,34,82,53,45,20,70,80,8,11,76,74,77,29,37,90,34,70,12,5]
[97,75,35,22,9,1,59,15,13,98,70,70,50,4,96,56,23,94,31,4]];
DueDate = [1188,1132,756,1109,934];

//Taillard_5_20_7
PTime = [[92,2,2,73,38,28,77,6,51,15,1,23,99,21,26,21,51,91,4,88]
[39,20,36,65,34,25,44,29,20,91,95,57,39,1,81,40,63,99,97,45]
[93,64,12,19,22,41,55,11,4,1,39,3,30,57,68,28,45,54,98,96]
[37,92,15,12,58,34,49,36,90,4,90,66,2,4,14,93,51,10,61,45]
[77,29,95,39,67,52,72,10,50,31,53,80,75,94,69,82,39,96,95,27]];
DueDate = [1598,1151,1505,863,1343];

//Taillard_5_20_8
PTime = [[21,8,61,62,67,28,30,70,92,31,26,65,13,6,24,49,73,68,31,25]
[83,14,55,23,86,68,70,76,34,12,45,58,60,28,55,97,92,30,32,62]
[42,64,47,35,75,29,29,4,85,48,24,33,72,20,60,15,53,12,14,30]
[92,66,28,62,57,53,46,58,69,26,86,10,64,37,83,8,41,13,53,36]
[45,68,33,43,34,53,25,53,86,55,56,80,83,58,3,63,33,58,4,41]];
DueDate = [1242,1267,791,1217,1066];

//Taillard_5_20_9
PTime = [[96,36,65,13,34,75,38,32,10,70,74,98,30,12,93,73,45,69,98,96]
[72,37,50,17,3,88,29,3,43,50,12,17,18,14,92,61,43,90,41,38]
[80,68,75,89,55,28,93,33,28,43,88,25,94,27,35,38,7,5,63,73]
[99,74,28,14,95,65,99,36,39,28,91,36,41,51,97,46,15,25,56,99]
[80,39,74,40,65,65,39,18,91,48,40,73,27,98,37,65,80,38,85,1]];
DueDate = [1157,1195,1268,1559,1405];

//Taillard_5_20_10
PTime = [[56,41,82,67,33,35,43,8,4,78,44,71,87,5,21,24,39,35,85,52]
[21,10,14,16,71,25,68,15,45,58,93,27,66,59,4,88,38,97,7,21]
[11,21,16,43,68,42,17,29,3,92,60,20,43,3,17,45,83,94,50,80]
[9,97,9,3,94,44,89,13,14,5,61,43,31,13,12,52,28,51,13,92]
[27,88,29,26,24,91,27,42,80,75,12,28,8,14,90,32,84,98,83,70]];
DueDate = [1314,1170,837,929,1131];
```