# Flexible Job-shop Scheduling Problem with Sequencing Flexibility: Mathematical Models and Solution Algorithms

By

Alejandro Vital Soto

A Dissertation
Submitted to the Faculty of Graduate Studies
through the Industrial and Manufacturing Systems Engineering Graduate Program
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy at the
University of Windsor

Windsor, Ontario, Canada

# Flexible Job-shop Scheduling Problem with Sequencing Flexibility: Mathematical Models and Solution Algorithms

by

Alejandro Vital Soto

APPROVED BY:

F. M. Defersha, External Examiner
University of Guelph

B. Chaouch
Odette School of Business

J. Urbanic
Department of Mechanical, Automotive and Materials Engineering

M. Wang
Department of Mechanical, Automotive and Materials Engineering

F. Baki, Co-Advisor
Odette School of Business

A. Azab, Co-Advisor
Department of Mechanical, Automotive and Materials Engineering

January 14, 2019

# DECLARATION OF CO-AUTHORSHIP / PREVIOUS PUBLICATION

## I. Co-Authorship

I hereby declare that this thesis incorporates material that is result of joint research of the author and his supervisors Prof. Ahmed Azab and Prof. Fazle Baki. Chapters 3, 4, 5, 6, 7 of the thesis was co-authored with Prof. Ahmed Azab and Prof. Fazle Baki. In all the cases, the key ideas, primary contributions, experimental design, data analysis, interpretation and writing were performed by the author; Prof. Ahmed Azab and Prof. Fazle Baki provided feedback on refinement of ideas and editing of the manuscript. This joint research has been submitted to Journals and Conferences that are listed below.

I am aware of the University of Windsor Senate Policy on Authorship, and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from Prof. Ahmed Azab and Prof. Fazle Baki to include the above material(s) in my thesis.

I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

## II. Declaration of Previous Publication

This thesis includes four original papers that have been previously published/submitted for publication in peer-reviewed journals and conferences, as follows:

| Thesis Chapter | Publication title/full citation | Publication status* |
|---|---|---|
| 3,6,7 | Vital Soto, A., A. Azab, and M. F. Baki. "A hybridized bacterial foraging optimization algorithm for the flexible job-shop scheduling problem with sequencing flexibility.” | Journal<br>(to be submitted) |

| | | |
|---|---|---|
| 4,7 | Vital Soto, A., A. Azab, and M. F. Baki. (To be submitted). "Mathematical modelling and multi-objective genetic algorithm for the dual-resource flexible job-shop scheduling problem with sequencing flexibility.” | Journal<br>(to be submitted) |
| 5,7 | Vital Soto, A., A. Azab, and M. F. Baki. "Multi-objective mathematical modelling for the flexible job-shop scheduling problem with process plan and sequencing flexibility.” | Journal<br>(to be submitted) |

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as a graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone’s copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

# ABSTRACT

Marketing strategists usually advocate increased product variety to attend better market demand. Furthermore, companies increasingly acquire more advanced manufacturing systems to take care of the increased product mix. Manufacturing resources with different capabilities give a competitive advantage to the industry. Proper management of the current productions resources is crucial for a thriving industry.

Flexible job shop scheduling problem (FJSP) is an extension of the classical Job-shop scheduling problem (JSP) where operations can be performed by a set of candidate capable machines. An extended version of the FJSP, entitled FJSP with sequencing flexibility (FJSPS), is studied in this work. The extension considers precedence between the operations in the form of a directed acyclic graph instead of sequential order. In this work, a mixed integer programming (MILP) formulation is presented. A single objective formulation to minimize the weighted tardiness for the FJSP with sequencing flexibility is proposed. A different objective to minimize makespan is also considered.

Due to the NP-hardness of the problem, a novel hybrid bacterial foraging optimization algorithm (HBFOA) is developed to tackle the FJSP with sequencing flexibility. It is inspired by the behaviour of the E. coli bacteria. It mimics the process to seek for food. The HBFOA is enhanced with simulated annealing (SA). The HBFOA has been packaged in the form of a decision support system (DSS). A case study of a small and medium-sized enterprise (SME) manufacturing industry is presented to validate the proposed HBFOA and MILP. Additional numerical experiments with instances provided by the literature are considered. The results demonstrate that the HBFOA outperformed the classical dispatching rules and the best integer solution of MILP when minimizing the weighted tardiness and offered comparable results for the makespan instances.

In this dissertation, another critical aspect has been studied. In the industry, skilled workers usually are able to operate a specific set of machines. Hence, managers need to decide the best operation assignments to machines and workers. However, they need also to balance the workload between workers while accomplishing the due dates. In this research, a multi-objective mathematical model that minimizes makespan, maximal worker workload and weighted tardiness is developed. This model is entitled dual-resource FJSP with sequencing flexibility (DRFJSPS). It covers both the machine assignment and also the worker selection.

Due to the intractability of the DRFJSPS, an elitist non-dominated sorting genetic algorithm (NSGA-II) is developed to solve this problem efficiently. The algorithm provides a set of Pareto-optimal solutions that the decision makers can use to evaluate the trade-offs of the conflicting objectives. New instances are introduced to demonstrate the applicability of the model and algorithm. A multi-random-start local search algorithm has been developed to assess the effectiveness of the adapted NSGA-II. The comparison of the solutions demonstrates that the modified NSGA-II provides a non-dominated efficient set in a reasonable time.

Finally, a situation where there are multiple process plans available for a specific job is considered. This scenario is useful to be able to react to the current status of the shop where unpredictable circumstances (machine breakdown, current product mix, due dates, demand, etc.) can be accurately tackled. The determination of the process plan also depends on its cost. For that, a balance between cost, and the accomplishment of due dates is required. A multi-objective mathematical model that minimizes makespan, total processing cost and weighted tardiness are proposed to determine the sequence and the process plan to be used. This model is entitled flexible job-shop scheduling problem with sequencing and process plan flexibility (FJSP-2F). New instances are generated to show the applicability of the model.

# DEDICATION

*I dedicate this dissertation to Jessica Olivares Aguila, Eugenio Alejandro Vital, Ana Maria Vital, Jorge Degante, Carmen Vital and God*.

# ACKNOWLEDGEMENT

The author would like to express his gratitude to his supervisors, Professor Ahmed Azab and Professor Fazle Baki, for their guidance, support, time and dedication. Their contributions and valuable feedback throughout all phases of this research have benefited and improved this dissertation.

The author also thanks to the committee members Dr. B. Chaouch, Dr. J. Urbanic, Dr. M. Wang, and Dr. F. M. Defersha for their constructive comments and discussions leading to significant improvements in this research.

Great thanks to the Production & Operations Management Research Lab (POM) colleagues for their valuable discussions.

I would like to express the most profound appreciation to Dr. Jessica Olivares Aguila to all her support, inspiration and love. I would like to express my deepest gratitude to my family Eugenio Alejandro Vital, Ana Maria Vital, Jorge Degante and Carmen Vital to provide me with the opportunity to reach this stage in my life.

The author acknowledges the funding provided by the National Council of Science and Technology in Mexico (CONACyT), and the Research Assistantship provided by Dr. Ahmed Azab and Dr. Fazle Baki. Also, I would like to thank the Department of Mechanical, Automotive & Materials Engineering for the Graduate Assistantship provided.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| ACO | Ant Colony Optimization |
| ANOVA | Analysis of Variance |
| BFOA | Bacterial Foraging Optimization Algorithm |
| DAG | Directed Acyclic Graph |
| DOE | Design of Experiments |
| DSS | Decision Support System |
| DRFJSPS | Dual-Resource Flexible Job Shop with Sequencing flexibility |
| EA | Evolutionary Algorithm |
| FJSP | Flexible Job Shop Scheduling Problem |
| FJSPS | Flexible Job Shop Scheduling Problem with Sequencing flexibility |
| FJSP-2F | Flexible Job Shop Scheduling Problem with sequencing and process plans Flexibility |
| FMS | Flexible Manufacturing System |
| GA | Genetic Algorithm |
| HA | Hybrid Algorithms |
| HBFOA | Hybrid Bacterial Foraging Optimization Algorithm |
| JSP | Job-shop Scheduling Problem |
| LEGA | LEarnable Genetic Architecture |
| MAV | Machine Assignment Vector |
| MILP | Mixed Integer Linear Programming |
| MOEA | Multi-objective Evolutionary Algorithm |
| MPS | Master Production Schedule |

| | |
|---|---|
| MRP | Material Requirement Planning |
| MRSLS | Multi Random Search |
| NRGA | Non-dominated Ranking Genetic Algorithm |
| NSGA-II | Elitist Non-dominated Sorting Genetic Algorithm |
| OSV | Operation Sequence Vector |
| SA | Simulated Annealing |
| SI | Swarm Intelligence |
| SME | Small Medium Enterprise |
| SPS | Single Point Search |
| TS | Tabu Search |
| VBA | Visual Basic for Applications |

# CHAPTER 1. INTRODUCTION

In this chapter, the motivation of this research is presented. Then, a brief overview of the production scheduling problem and how to tackle is provided. Machine scheduling under the bigger umbrella of production planning is outlined. Scheduling terminology is presented, and the base of the scheduling function importance is discussed. Moreover, an introduction to evolutionary algorithms and multi-objective optimization is given. Finally, the objectives and research plan of this research are being presented.

## 1.1 Motivation

Marketing strategists usually advocate increased product variety in order to attend better to market demand. Furthermore, companies often acquire more advanced manufacturing systems to take care of the increased product mix and unpredictable demand. Flexible manufacturing systems (FMS) is an example of the production environment that has been used to answer the needs of a turbulent market environment. This production system encompasses different levels of changeability. They have been implemented but are not fully optimized. Strategic scheduling of these complex systems would thus require decisions with conflicting objectives. Hence, methodologies to tackle these problems would be needed to foster a thriving industry.

The FJSP is an extension of the classical job-shop scheduling problem (JSP), where operations can be performed by a set of candidate capable machines (Brandimarte 1993).

Process planning is a significant function in the manufacturing process, and this is the process of selecting and sequencing manufacturing processes by achieving one or more objectives (Shen *et al.* 2006). A process plan is generated without information of the current manufacturing resources, and it lacks consideration of scheduling objectives. The process plan created could also not be feasible due to the present manufacturing scenario. Generally, some modifications of the process plan need to be done to reflect the updated shop floor status. Sequential executions of the process planning and scheduling results have been ineffective thus far. An integration of these two functions that take into account the objectives that both problems could optimize should be considered.

## 1.2 Production scheduling

The process to arrange a set of task over the time where capability, capacity and time constraints are considered is defined as the production scheduling problem (Lopez and Roubellat 2013). Production scheduling can be seen as a decision-making process that is used on a regular basis in many manufacturing and service industries (Pinedo 2005). In order to describe the importance of scheduling process, it is necessary to show how it fits under the bigger picture of the production function.

Production scheduling is a crucial element of the production function. The production function can be view as a hierarchical process (Nahmias 2009). The first step is to forecast the demand for aggregate sales. The forecast should be done over a predefined planning horizon. These data are the input to create an aggregate plan (aggregate production and workforce levels) which will be translated into the master production schedule (MPS).  The MPS results in specific production goals by product and time period.

One technique to reach the production goals of finished-goods inventory produced by the MPS is the materials requirement planning (MRP). The MRP breaks down the MPS into a detailed schedule of production of each component that comprises an end item. The result of the MRP is transformed into specific shop floor schedules that include a set of tasks and due dates related to those tasks. An illustration of this hierarchical process can be found in Figure 1.1.

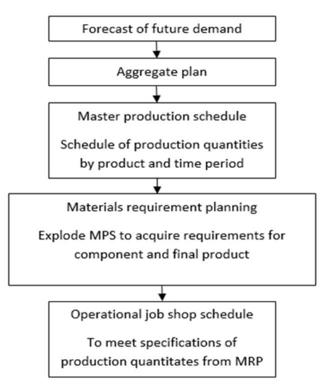



*Figure 1.1 Hierarchy of production decisions(adopted from Nahmias (2009)).*

Production scheduling is one of the critical production planning enablers to facilitate responsiveness and agility to achieve such flexibility. In many productions systems the scheduling function is usually performed by first line management, but over the years this function has become a more complex task to achieve. For this reason, much attention has been brought to the scheduling function.

The traditional job-shop scheduling problem (JSP) is defined as the suitable sequencing of a set of jobs that must be processed on different machines while satisfying a set of prescribed precedence constraints (Demir and Kürşat İşleyen 2013). The job-shop scheduling problem has been widely studied since the 1950s (Özgüven *et al.* 2010). JSP has been proved to be NP-hard (Garey *et al.* 1976).

This research work, review the different techniques that have been proposed over the years and establish a methodology to study an extension of the FJSP.

## 1.3 Scheduling framework

In scheduling theory, the number of jobs and machines are assumed to be finite. $n$ represents the number of jobs and the number of machines is define by $m$. Generally, the index $j$ is used to refer to a job and the index $i$ is employed to denote a machine. The processing of a job $J_j$ on machine $M_i$ is called an operation $o_{ij}$.

Scheduling problems are usually described by a triplet $\alpha \mid \beta \mid \gamma$ (Pinedo 2005). The first entry $\alpha$ represents the machine environment. The second entry $\beta$ provides information of processing characteristics and constraints. It might comprise no entry at all or multiple entries. The last entry $\gamma$ describes the objective to be minimized.

The most basic machine environments $\alpha$ are described by Pinedo (2005) as follows:

- Parallel machines (*Pm*): There are m identical machines in parallel. Each job needs one and only one of these machines.
- Flow shop (*Fm*): All the jobs have to be processed on each one of the machines, and each job has to follow the same route.
- Flexible flow shop (*FFc*): A flexible flow shop is a generalization of the flow shop and the parallel machine environments. Instead of $m$ machine in series, there are $c$ stages in series with a number of identical machines in parallel at each stage. Each job has to be

processed first at Stage 1, then Stage 2, and so on. A stage functions as a bank of parallel machines; at each stage, job $j$ requires processing on only one machine and any machine can perform the job.

- Job shop (*Jm*): In this case, each job has its own predefined route to follow through the machines.
- Flexible job shop (*FJc*): A flexible job shop is a generalization and the parallel machine environment. Each job has its own route to follow through the shop. Each job is processed only by specific machines.
- Open shop (*Om*): Each job has to be processed on each machine. However, some of these processing time may be zero. There are no restrictions with regard to the routing of each job through the machine environment. The scheduler is allowed to determine a route for each job, and different jobs may have different routes.

The typical processing restrictions and constraints in the second entry $\beta$ are provides as follow:

- Sequence-dependent setup times ($s_{jk}$): The $s_{jk}$ represents the sequence dependent setup time between jobs $j$ and $k$.
- Precedence constraints (*prec*): In this case, one or more jobs need to be completed before another job is allowed to start its processing.
- Machine eligibility restrictions ($M_j$): In this scenario, some of the machine are capable of processing job $j$. $M_j$ symbolizes a set of capable machines that can process the job $j$.

For the last entry $\gamma$, the most common objectives used are:

- Makespan ($C_{max}$): Completion time of the last job on the system.
- Maximum lateness ($L_{max}$): The lateness of a job is define by $L_j = C_j - d_j$. $C_j$ represents the completion time of job $j$ and $d_j$ is the due date of a job. $L_j > 0$ when a job is completed late and $L_j < 0$ when is completed early. The $L_{max}$ measures the worst violations of the due dates.
- Weighted tardiness ($\sum W_j T_j$): $T_j$ is defined as $T_j = max(C_j - d_j, 0)$ and $W_j$ is the weight (penalty) of each job. The weighted tardiness is defined as $\sum W_j T_j$.

## 1.4 Flexibility in manufacturing systems

Review of the literature identifies at least ten types of manufacturing systems flexibilities (ElMaraghy 2005). These are:

1. Machine flexibility: Various operations performed without set-up change,
2. Material handling flexibility: Number of used paths / total number of possible paths between all machines,
3. Operation Flexibility: Number of different processing plans available for part fabrication,
4. Process Flexibility: Set of part types that can be produced without major set-up changes, i.e. part-mix flexibility,
5. Product Flexibility: Ease (time and cost) of introducing products into an existing product mix. It contributes to agility,
6. Routing Flexibility: Number of feasible routes of all part types/Number of part types,
7. Volume Flexibility: The ability to vary production volume profitably within production capacity,
8. Expansion Flexibility: Ease (effort and cost) of augmenting capacity and/or capability, when needed, through physical changes to the system,
9. Control Program Flexibility: The ability of a system to run virtually uninterrupted (e.g. during the second and third shifts) due to the availability of intelligent machine and system control software,
10. Production Flexibility: Number of all part types that can be produced without adding major capital equipment.

## 1.5 Introduction to the flexible job-shop problem

The manufacturing companies, in an effort to react to the unpredictable demand, acquire more manufacturing equipment. The new machines usually have more advanced capabilities. For instance, the new equipment may process operations faster, and it could handle larger capacities. Generally, companies maintain the two kinds of manufacturing resources. In this manner, in case of a breakdown or scheduled maintenance, the old resource can be used to keep the production running. Furthermore, the old resource can be used as a method to balance the production when one of the resources is overloaded. With this scenario, companies end up with a combination of new and old equipment. These new production environments are a new challenge for managers when they have to execute the production function.

In response to the new productions environments, an extension of the JSP named flexible job shop scheduling problem (FJSP) has been developed. The FJSP was first presented by Brucker and Schlie (1990), it allows operations to be processed on any of the available flexible machines, the machines are not necessarily identical. The flexibility in an FJSP can be either total or partial (Kacem *et al.* 2002b). Total flexibility is defined when all the operations can be processed on any of the machines, and partial flexibility is when some operations can be processed on the available machines.

The decision involved in the FJSP can be categorized into sub-problems (Brandimarte 1993):

- Routing sub-problem: The assignment of each operation to a machine out of a set of capable machines is performed.
- Scheduling sub-problem: It is as well known as the sequencing problem. This type of problem selects the operation sequence of all the jobs.

Pinedo (2005) represents a FJSP with total flexibility with $FJc|C_{max}$ and the case of partial flexibility $FJc|M_j|C_{max}$ . The FJSP belongs to the NP-hard class (Demir and Kürşat İşleyen 2013).

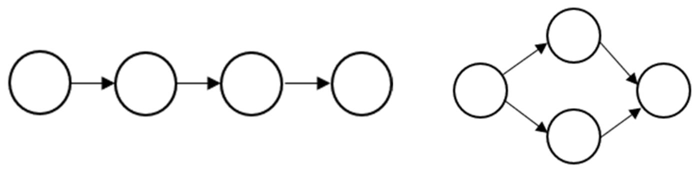
Due to the intractable complexity of the FJSP, two elementary approaches have been developed (Brandimarte 1993):

- Concurrent approach: The general idea of this approach is to tackle the routing and scheduling problem together. In other words, the problems are integrated.
- Hierarchical approach: This approach has been widely used, and it explodes the main problem, and it handles each problem (routing and scheduling) in a sequential manner.

### 1.5.1 Sequencing flexibility

The FJSP assumes sequential order precedence relationships. For some production environments, the precedence relationship between the operations is not always given in a sequential order. It can be determined by a directed acyclic graph (DAG). This situation is illustrated in Figure 1.2. The decisions involved in this extension of the FJSP, consist of the machine assignment and the assurance of the precedence constraints. Hence, the FJSP with sequencing flexibility permits the precedence between the operation to be given by an arbitrary DAG instead of sequential order

(Birgin *et al.* 2015). This research considers the mentioned extension, that is a typical example in industries such as printing (Birgin *et al.* 2015; Birgin *et al.* 2014; Vilcot and Billaut 2008), glass (Alvarez-Valdés *et al.* 2005), mould (Gan and Lee 2002) and the cutting tool industry. For instance, in the cutting tool industry, the FJSP with sequencing flexibility is tackled. Specifically, during the honing and heat treat operations. The execution of these two operations does not present a relationship. That is, honing could be performed after or before heat treat without any problem in the production process. Nevertheless, this sequence flexibility gives the advantage to managing the production resources better. The manager could decide on the process to be performed based on the machines' availability.



(a) (b)

*Figure 1.2 Precedence relationship of a single job (a) Sequential order (b) DAG.*

## 1.6 Evolutionary algorithms

Metaheuristics based on natural selection are known as Evolutionary Algorithms (EA). These solution methods are based on a population of solutions that are generated, selected, combined and updated. There are three types of EA: Genetic algorithms (GA), Evolutionary programming (EP) and Evolutionary Strategies (ES). EAs have proven very successful in practical applications, they are highly flexible and can be configured to address any optimization task. EA uses three evolutionary operators (mutation, recombination and selection) to generate better solutions. Mutation and recombination are used to create a new solution. Then, above-average individuals in the population are selected (reproduced) to become members of the next generation (Coello *et al.* 2007).

### 1.6.1 Genetic algorithms

GA are search methods based on principles of natural selection and genetics. GA have been widely used to generate high-quality solutions to optimization problems. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation, as represented in Figure 1.3.

GA encode the decision variables of a problem in a chromosomal manner and use a fitness measure for discriminating good solutions from bad ones. GA is initialized with a set of random solutions called population. Each solution in the set is named a chromosome that represents a problem solution. The chromosomes are evaluated with a fitness function to determine how well the solutions solve the problem. This function is used to select parents that are used to create new generations. Two processes are employed for the formation of a new solution, usually called offspring. The first process is the crossover or recombination, with the selected characteristics from the parents creates a new solution. The second process is the mutation in which new genetic material is introduced. After the mutation process, the solutions are evaluated with the fitness function. The top scoring members of the population are chosen. The algorithm continues running up to it reaches a specified number of runs or a fitness threshold. After several generations, the algorithm converges to the best solution (Burke and Kendall 2005).

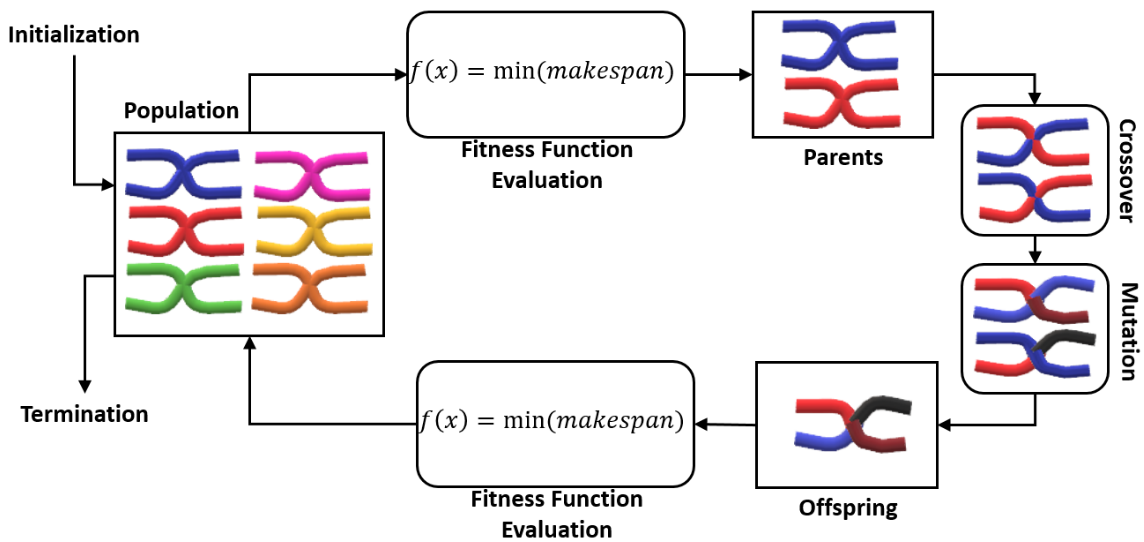



*Figure 1.3 Genetic algorithm representation.*

## 1.7 Multi-objective optimization

Decision making usually involves conflicting objectives. In real problems, no single solution simultaneously optimize those conflicting objectives. Contrary, there is a set of trade-off optimal solutions commonly known as Pareto-optimal. It is important to find as many Pareto-optimal solutions, so the user would be in a better position to make a choice when many such trade-off solutions are unveiled (Burke and Kendall 2005).

For multi-objective optimization, multiple fitness functions are used to provide a set of optimal solutions. This set of solutions contains elements that are equally optimal as none of the objective functions can be improved in value without degrading some of the other objective values. That is, no solution dominates any other solution in the frontier as presented in Figure 1.4. These solutions are usually called non-dominated solutions or Pareto-optimal solutions. Several methods for multi-objective optimization have been developed in the literature, from the classic methods as the weighted- sum approach and $\epsilon$-constraint method to the elaborated evolutionary algorithms (Burke and Kendall 2005).

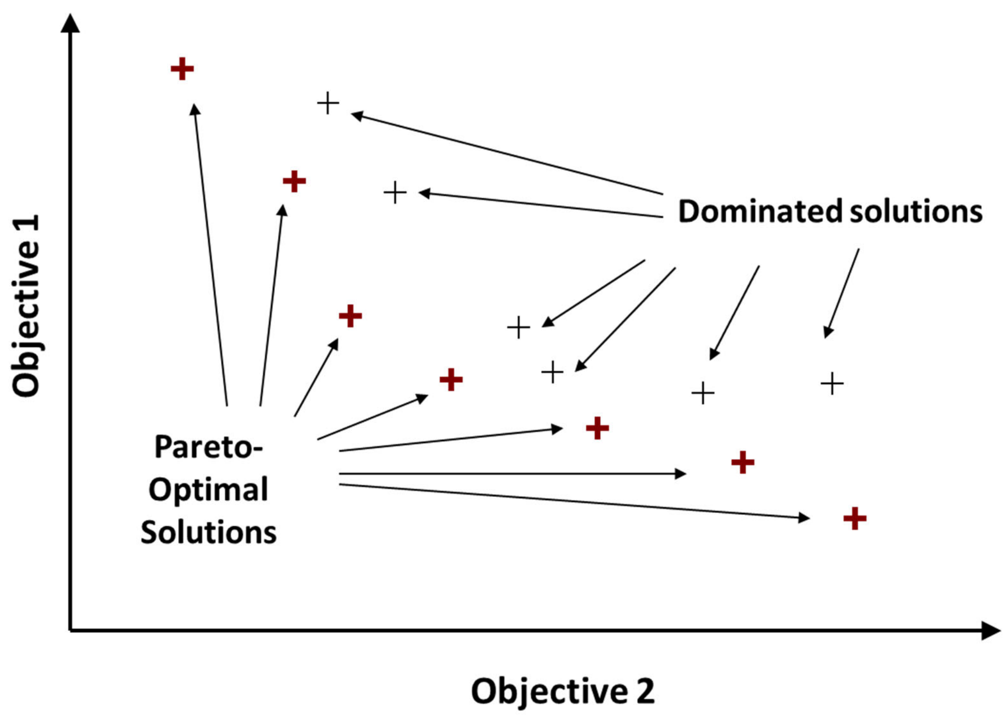



*Figure 1.4 Pareto frontier representation for the minimization of two objectives.*

### 1.7.1 Weighted-sum approach to multi-objective optimization

The simplest method for multi-objective optimization is the weighted-sum approach. This method scalarizes a set of objectives into a single objective by multiplying each objective with weights that reflect the decision maker's preference. As the determination of specific values of those weights is subjective, this method could yield only an efficient but not optimum solution. Additionally, the weight vector corresponds to a pre-destinated optimal solution on the Pareto frontier. Hence, changing the weight vector, a different Pareto-optimal point can be obtained (Burke and Kendall 2005).

### 1.7.2 Elitist non-dominated sorting GA (NSGA-II)

Multi-objective evolutionary algorithms (MOEAs) have been used in the literature due to their ability to find multiple Pareto-optimal solutions in one single simulation run (Deb *et al.* 2002). The nondominated sorting genetic algorithm (NSGA) was proposed by (Srinivas and Deb 1994). This algorithm has been criticized due to its high computational complexity, the lack of elitism approach and the need for specifying a sharing parameter (Deb *et al.* 2000). Hence, to solve the algorithms pitfalls, the elitist non-dominated sorting GA (NSGA-II) was introduced by Deb *et al.* (2002).

The NSGA-II for multiple Pareto-optimal solutions in a multi-objective optimization has three essential features: It uses an elitist principle, it uses an explicit diversity preserving mechanism, and it emphasizes the non-dominated solutions (Burke and Kendall 2005). NSGA-II tends to spread quickly and appropriately when a certain non-dominated region is found.

## 1.8 Thesis statement

The integration of dual-resource approach, sequencing flexibility, process plan flexibility and process plan cost for the FJSP and the development of mathematical models and new metaheuristics yields a better scheduling solution that requires less time and cost.

Mathematical programming and metaheuristics, specifically the hybrid bacterial foraging optimization algorithm and Elitist Non-dominated Sorting GA, can solve the FJSP efficiently with sequencing flexibility.

## 1.9 Research objectives

The overall objectives of the proposed work are as follow:

1. Develop an alternative way of formulating the FJSP with sequencing flexibility.

2. Develop and implement a metaheuristic to solve larger instances of the extended version of the FJSP.
3. Formulate a mathematical model for the dual-resource FJSP with sequencing flexibility.
4. Develop and implement a metaheuristic to solve larger instances of the dual-resource FJSP with sequencing flexibility.
5. Develop a mathematical model for the FJSP with sequencing and process plan flexibility and incorporate the processing cost.
6. Provide instances with these features to the literature.

## 1.10 Research plan

The proposed methodology is comprised of three stages. The first one is called "A MILP formulation for the FJSP with sequencing flexibility." The second one is named "A mathematical model and metaheuristic for the dual-resource FJSP with sequencing flexibility." The third stage is entitled "Mathematical model for the FJSP with sequencing and process plan flexibility," which is to be started after finish the second stage. The steps for each stage are shown in Figure 1.5, Figure 1.6 and Figure 1.7, respectively.

The first step of Stage 1 develops a single objective mathematical model. The model will incorporate the sequencing flexibility condition. An MILP formulation technique that minimizes the weighted tardiness or makespan will be used. The mathematical formulation is a crucial step to understand the structure of the problem and to develop an effective metaheuristic (Unlu and Mason 2010). The next step is related to the metaheuristic implementation. A search will be done for a proper metaheuristic that can be used as an alternative for the most popular metaheuristics. Special attention has to be taken to find a metaheuristic that takes into account the different characteristics of the FJSP. Most of the metaheuristics employ parameters that need to be set based on the problem nature. A technique that can be used to set these parameters is design of experiments, specifically a $2^k$ factorial design. The $2^k$ factorial design is suitable in factor screening process (Montgomery 2008). This design works with the smallest number of runs where *k* factors can be studied. This technique has been used for several authors to set up metaheuristics parameters, an example is provided in Vital Soto *et al.* (2017). This type of design will be helpful to detect the significant main factors and interactions. After this, a regression model can be generated to find the parameter settings (main factor and/or interactions) that yield the best expected response. Further analysis could be studied, as a general factorial design or response surface methodology, in case of the need of more specific parameters values. The last two steps of stage 1 are based on the efficiency evaluation of the mathematical model against alternative formulations and to perform experiments with well-known instances for FJSP.

For stage 2, the same methodology explained for stage 1 will be carried out with the addition of the development of instances with worker selection. Some of the instances will be taken from the literature and modified to suit the current problem. The mathematical formulations will take into account a multi-objective optimization with objective functions that can capture current manufacturing environments as minimization of the weighted tardiness, maximal worker workload and makespan.

For stage 3, a new mathematical formulation will be developed to consider process plan and sequencing flexibility. The multi-objective will study the minimization of processing cost, makespan and weighted tardiness. Several instances will be created to test the performance of the mathematical formulation.

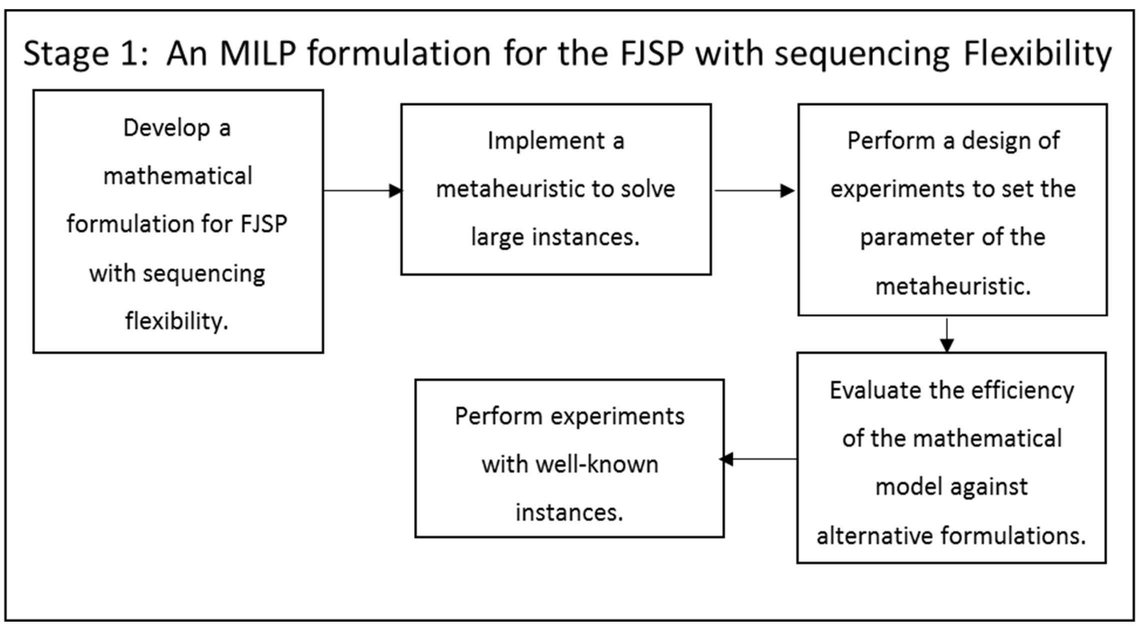



*Figure 1.5 Stage 1 for the proposed methodology.*

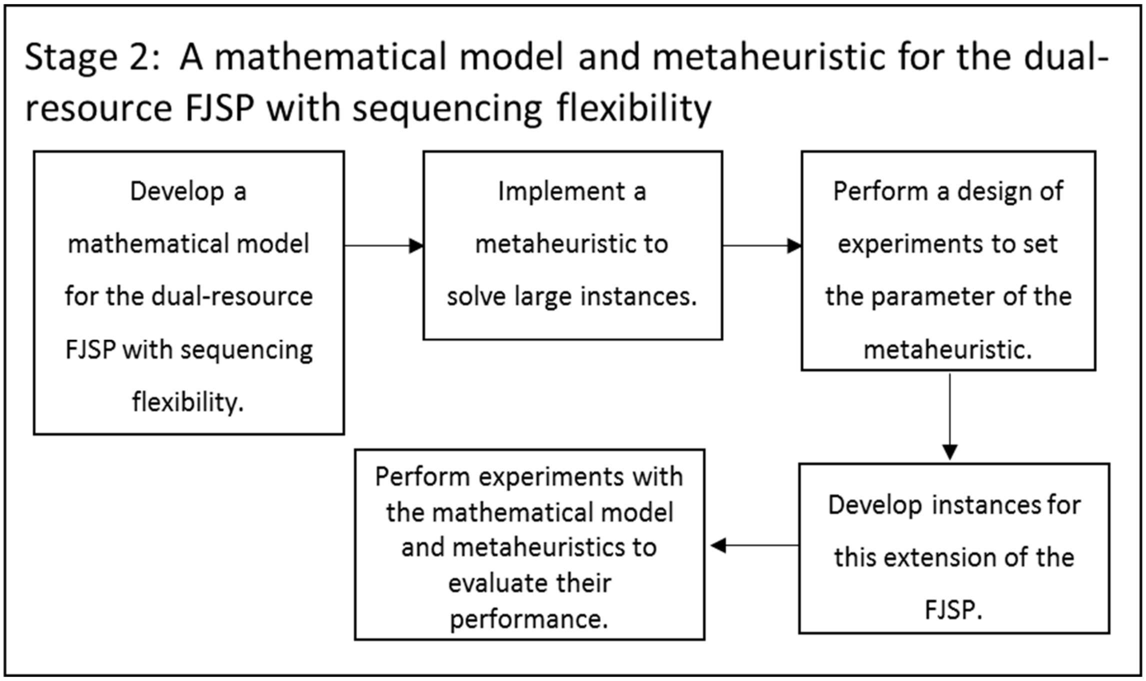


*Figure 1.6 Stage 2 for the proposed methodology.*

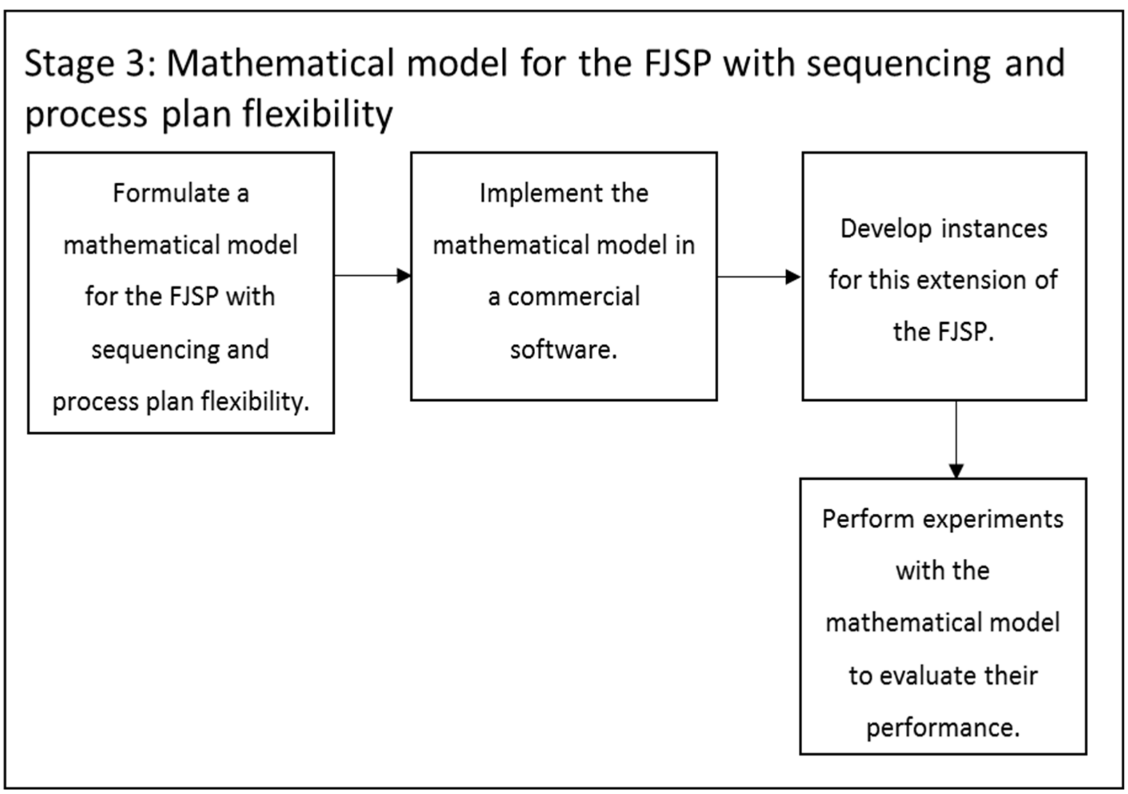


*Figure 1.7 Stage 3 for the proposed methodology.*

# CHAPTER 2. LITERATURE REVIEW

The FJSP has been studied during the last 28 years. This chapter provides the most relevant literature for the FJSP. It emphasizes to classify the researched performed in two categories: mathematical formulations and metaheuristics. Then, this review identifies gaps in the literature that lead to the formulation of this research.

## 2.1 Mathematical formulations for the flexible job-shop scheduling problem

Mathematical models for production scheduling were first developed around 1960 (Özgüven *et al.* 2010). Wagner (1959), Manne (1960) and Bowman (1959) presented mathematical formulations that categorized the different mathematical formulation. These categories are identified as position-based, sequence-based and time-based.

### 2.1.1 Position-based model

Wagner (1959) proposed the first position based mathematical model for the JSP. The base of this formulation assumes that each machine has some positions or slots. This approach considers a binary variable that assigns an operation to a position on a machine. For the FJSP, Lee *et al.* (2002) were the first to introduce a mathematical formulation to minimize makespan considering process plan flexibility, outsourcing, and due dates. Later, Fattahi *et al.* (2007) presented a mixed integer linear programming (MILP) to minimize makespan and testing problems (small and medium-large size problems). With their formulation, they were able to solve small size problems and for medium-large size problem upper and lower bounds were reported. Similarly, Fattahi *et al.* (2009) proposed an MILP allowing overlapping in operation when minimizing makespan. The authors presented the situation where each job has a demand for more than one, a case usual in petrochemical industries.

Roshanaei *et al.* (2013) studied two MILP models for the FJSP. Minimization of makespan was selected as the optimization criterion. They analyzed their models by examining data proposed by Fattahi *et al.* (2007), and they studied an industrial problem in a mould and die-making shop as well. The performance of their model was assessed by counting the binary, continuous variables and the number of constraints. The author introduced another performances measures as size complexity, computational time and the quality of schedules generated. Their position based model outperformed the mathematical formulation presented by Fattahi *et al.* (2007) based on the performance measures proposed.

Abdi Khalife *et al.* (2010) developed a multi-objective FJSP with overlapping in operations. In their study, they considered a multi-objective function that comprises makespan, critical machine work loading time, and total machine work loading time.

### 2.1.2 Sequence-based model

Manne (1960) was the pioneer to propose a sequence based mathematical model for JSP. The model bases his functionality by precedence binary variables that define a sequence of operations that are assigned to the same machine. Disjunctive Constraints are presented in this formulation.

For the FJSP, Kim and Egbelu (1999) developed an MILP model involving multiple process plan. The model determines both process plan and a schedule that minimize makespan. Another essential mathematical formulation is the one studied by Low and Wu (2001). They proposed a 0-1 integer programming model with the objective of minimizing the total tardiness in a flexible manufacturing system considering setup time. The setups were assumed to be sequence independent. They presented a linearization of their quadratic constraints.

Zhang *et al.* (2012) showed a model for the FMS with transportation and bounded processing time to minimize makespan and storage. In this study, all jobs need to be transported to be processed by a machine, the transportation can be done by any transport and the loading/unloading time is machine dependent.

Özgüven *et al.* (2010) established two MILP models. The first model deals with FJSP minimizing makespan, and it is compared with the model presented by Fattahi *et al.* (2007). The second model studied the FJSP by adding process plan flexibility.

Other considerations have been formulated for the FJSP, Gao, Gen*, et al.* (2006) modelled a situation when the availability of the machines is compromised due to maintenance, pre-schedules, etc. The authors studied the FJSP with non-fixed availability constraints. In this study, the model minimizes three objectives defined as makespan, maximal machine workload and total workload of the machines.

Imanipour (2006) proposed an extension to the FJSP considering process planning and sequence flexibility. Therefore, a mixed-integer nonlinear programming is presented with sequence dependent setup times between jobs. The purpose is to find the best operation sequence, best machine, and best operation schedule while trying to minimize makespan. Instances of full flexible job shops problems were used. Additionally, setup times were assumed to be the same for all machines.

A new MILP model to minimize makespan was introduced by Birgin *et al.* (2014) as an extension of the FJSP problem. This model allows an arbitrary precedence relation over the set of operations rather than a sequential order. The authors developed a new set of instances based on the printing industry case. Some experiments were carried out; the results showed that the model generates better results than the model presented by Özgüven *et al.* (2010).

### 2.1.3 Time-based model

Bowman (1959) introduced this mathematical approach for the JSP. This approach considers time indexed binary variables that assign operations to time periods of capable machines. This formulation has not been popular in the literature. For Instance, Gomes* *et al.* (2005) used an integer linear programming (ILP) formulation to model the FJSP in make-to-order industries that allow job re-circulations. Another example of this kind of formulation, it is the work presented by Thomalla (2001). A model for scheduling jobs in a just-in-time environment was introduced. The model was formulated as a discrete-time integer optimization. This model minimizes the weighted quadratic tardiness and considers parallel machines with possible different processing time and efficiency. Moreover, it considers non-preemptive scheduling of jobs. For that, the jobs have the earliest start time, due date and a penalty that is applied when the due date is not met. This penalty increases as the tardiness increases. The model was tested in small examples with a maximum of 6 jobs, ten operations, and six machines.

## 2.2 Metaheuristic methods

Due to the NP-hard characteristic of the FJSP, exact approaches are not able to find a solution for large-scale FJSP instances (Pezzella *et al.* 2008). For this reason, metaheuristic methods have been used to solve the FJSP. Many metaheuristics have been presented in the literature such as evolutionary algorithms (EA), swarm intelligence (SI), single point search (SPS), hybrid algorithms (HA), etc. A review of the most relevant works is shown as follow.

### 2.2.1 Evolutionary algorithms

Pezzella *et al.* (2008) presented a genetic algorithm (GA) for the FJSP. The effectiveness of GA relies on the incorporation of different strategies for selection, reproduction, and formation of the initial population. The methodology was tested on 178 problems from literature, and the results showed that the solution quality is comparable to tabu search.

Another research dedicated to solving the FJSP using GA is the one presented by De Giovanni and Pezzella (2010). They considered the minimization of the global makespan overall flexible manufacturing units (FMU) in a distributed flexible job-shop scheduling problem. The improved GA showed to be very robust and be able to find optimal or near-optimal solutions with reduced computational times.

Ahmadi *et al.* (2016) developed a multi-objective approach for FJSP considering random machine breakdown. Stability and makespan were considered as optimization variables. Two algorithms, non-dominated ranking genetic algorithm (NRGA) and NGSA-II, and simulation were applied to evaluate the state and condition of the machine breakdowns. As a result, a set of Pareto solutions was obtained.

LEarnable Genetic Architecture (LEGA) was proposed by Ho *et al.* (2007). This approach integrates evolution and learning within a random search. Similarly, Demir and İşleyen (2014) proposed a GA for the FJSP with overlapping in operations. Driss *et al.* (2015) developed a new GA that uses a new chromosome representation and implements different strategies for crossover and mutation.

Recently, a new hybrid genetic algorithm (NHGA) has been developed by Gong *et al.* (2018) for the double FJSP considering worker selection. The NHGA presented a new chromosome encoding method, crossover and mutation operators. A multi-objective mathematical formulation was developed where the researchers studied three objectives: minimize makespan, minimize the total worker cost and the maximization of the green production indicators.

### 2.2.2 Swarm intelligence

Swam intelligence has been applied to tackle the FJSP. These metaheuristics include ant colony optimization (ACO) algorithm, artificial bee colony (ABC) algorithm, particle swarm optimization (PSO) algorithm, etc.

Rossi and Dini (2007) investigated the FJSP with routing flexibility, sequence-dependent setup times and transportation time using ACO algorithm. Additionally, parallel machines and operation lag times were considered to tailor the ant colony operators for improving the solution quality. The new methodology was proven to be effective to solve FMS scheduling with high variety.

Xing *et al.* (2010) developed a knowledge-based ant colony optimization (KBACO) for the FJSP. It uses the ant colony heuristic to search and identify an optimal solution, and the knowledge model learns from the optimization and guides the current heuristic searching. The experiments showed that this algorithm obtains near-optimal solutions with quick computational speed.

Huang *et al.* (2013) focused on the FJSSP considering due window and sequence-dependent setup times. They developed a two-pheromone ant colony optimization (2PH-ACO). This is an improved ACO that uses two pheromones.

ABC algorithm has been used by Wang *et al.* (2012). They proposed particular encoding and decoding schemes and effective operator to minimize makespan. Gao *et al.* (2015) presented a two-stage ABC algorithm for scheduling and re-scheduling with job insertion.

Gao, Peng*, et al.* (2006) defined a general particle swarm optimization (GPSO) algorithm to tackle the traditional limitation of traditional PSO to solve the FJSP. The omitted the concrete velocity-displacement updating method in the traditional PSO.

Recently, researchers developed a knowledge-guided fruit fly optimization algorithm (KGFOA) to solve the dual-resource constrained FJSP (Zheng and Wang 2016). The problem deals with the job sequence, machine assignment and worker assignment to minimize makespan. The KGFOA proposed a new encoding scheme and two types of permutation-based search operators.

### 2.2.3 Single point search

Different SPS metaheuristics are available in the literature, the effectiveness of them relies on the design of neighbourhood structures. Methods like Tabu Search (TS), simulated annealing (SA) and variable neighbourhood search (VNS) are considered in this category.

TS is a metaheuristic that incorporates adaptive memory and responsiveness exploration. It has been proven to be the most effective method for solving the scheduling problem. Brandimarte (1993) was the first to develop a hierarchical algorithm that decomposes the FJSP into a job shop scheduling sub-problem and a routing sub-problem. As a result, two interaction schemes between the sub-problems were used. Additionally, he used dispatching rules to solve the assignment problem and TS to focus on the resulting job shop sub-problems. Hurink *et al.* (1994) presented a TS that considers reassignment and rescheduling as two different moves. Mastrolilli and Gambardella (2000) introduced two new neighbourhood structures for solving the FJSSP. They reduced the set of possible neighbours to a subset that always includes the neighbour with the lowest makespan.

A well-known metaheuristic used to solve several optimization problems is the variable neighbourhood search (VNS). In the paper presented by Amiri *et al.* (2010), neighbourhood solutions are generated using six neighbourhoods related to the sequencing and assignment problem to solve the FJSP. Similarly, Yazdani *et al.* (2010) developed a parallel VNS to solve the FJSP. Makespan minimization was considered, and parallelization was carried out to increase the exploration in the search space. Neighbourhood structures related to sequencing, assignment and the combination of both were used to search the solution space.

SA models the process of heating material and then slowly lowering the temperature to decrease defects. SA models have been used to solve highly nonlinear problems. In the FJSP, SA has been used by Loukil *et al.* (2007). They used a multi-objective SA that considers makespan, the mean completion time, the maximal tardiness and the mean tardiness. As a result, the SA algorithm provides a set of efficient schedules that incorporates batch production, two-step process and overlaps of the processing periods of two successive operations. Fattahi *et al.* (2009) proposed a mathematical model that includes SA and TS to minimize makespan in an FJSSP with overlapping operations. Consequently, they developed six different algorithms for the FJSSP.

### 2.2.4 Hybrid algorithms

To take advantage of the strengths of the metaheuristic methods, some researchers have developed hybrid techniques to construct an effectively hybridized algorithm (HA). Some researchers have used HA to solve the FJSSP. For instance, in the paper presented by Kacem *et al.* (2002b), a Pareto approach based on hybridization of Fuzzy Logic (FL) and Evolutionary Algorithms (EAs) was introduced. This approach takes advantage of the knowledge representation capabilities of FL and the adaptive capacities of EAs. Another work dedicated to HA is the one presented by Xia and Wu (2005). They implemented a hybrid approach for the multi-objective FJSSP using particle swarm optimization and SA. The resulting method was experimentally proven to be effective for the problem on a large scale. In the research developed by Birgin *et al.* (2015), a list scheduling algorithm and a filtered beam search method are hybridized to solve the extended FJSP. González *et al.* (2015) developed a new algorithm that includes scatter search and path relinking was presented for the FJSP. They proposed neighbourhood structures that are embedded into a scatter search algorithm which uses TS and path relinking in its core. The effectiveness of this algorithm relies on the diversification provided by the different heuristics. Li and Gao (2016) hybridized the GA and TS to minimize makespan in the FJSP. The proposed algorithm takes advantage of the GA to perform global exploration and the TS to perform local exploitation. The experimental results showed that a significant improvement for solving FJSP was achieved.

## 2.3 Summary and research gaps

The FJSP has been studied during the last 28 years. Different techniques as mathematical formulations, metaheuristic methods, and hybrid techniques have been developed to solve the FJSP, and as well new considerations have been added to the simple FJSP.

In Table 2.1, a synthesis matrix is presented. Sequence-based models are the most studied mathematical formulations. Most of the work reviewed have used the minimization of makespan as the objective function. Other performance measures as tardiness, total workload, etc., have been less studied in the literature. Single objective optimization problems have attracted more the attention of the researchers. GAs is the most popular metaheuristic to solve the FJSP. Hybrid algorithms have widely used to deal with the FSJP. Most of the researchers have studied the simple FJSP. However, some authors have considered other features of the problem as flexible process plans, setup time, overlapping in operations, etc.

From the literature, it is evident that the research on FJSP has tried to incorporate more conditions that have been neglected or assumed with the aim of giving a more realistic situation in a typical environment. It is important to note that an extension that is not thoroughly researched is the FJSP with sequencing flexibility when the precedence between the operations are given by a directed acyclic graph instead of a sequential order (an example is provided in Figure 1.2). Practical examples can be found in the printing industry (Birgin *et al.* 2015) where assembly and disassembly operations are part of the process, manufacturing industries where the operation sequencing is performed satisfying all given precedence.

*Table 2.1 Synthesis matrix.*

| Article | Mathematical Model | Algorithm | Objective Function | Single/Multi Objective | Remarks |
|---|---|---|---|---|---|
| Lee *et al.* (2002) | Position-based | GA | Min. Cmax | Single | FJSP with process plan flexibility, outsourcing, and due dates |
| Fattahi *et al.* (2007) | Position-based | Integrated and hierarchical SA and TS | Min. Cmax | Single | FJSP instances (small and medium-large size problems) |
| Fattahi *et al.* (2009) | Position-based | SA | Min. Cmax | Single | FJSP with overlapping in operations |
| Roshanaei *et al.* (2013) | Position-based Sequence-based | Hybrid Artificial Immune and SA | Min. Cmax | Single | FJSP for an  industrial problem in a mould and die-making shop |
| Abdi Khalife *et al.* (2010) | Position-based | SA | Min. Cmax, Total machine work loading time and critical machine work loading time | Multiple | FJSP with overlapping in operations |
| Kim and Egbelu (1999) | Sequence-based | Preprocessing algorithm and iterative algorithm | Min. Cmax | Single | FJSP with multiple process plans |
| Low and Wu (2001) | Sequence-based | SA based heuristic | Min. Total Tardiness | Single | FJSP with sequence independent setup time. |
| Zhang *et al.* (2012) | Sequence-based | GA with TS | Min. Cmax and bounded processing times | Multiple | FJSP with transportation constraints and bounded processing times |
| Gong *et al.* (2018) | Sequence-based | NHGA | Min. Cmax, maximal total worker cost and Max. green production indicator | Multiple | FJSP with worker selection |
| Özgüven *et al.* (2010) | Sequence-based | - | Min. Cmax | Single | FJSP with process plan selection |
| Gao, Gen*, et al.* (2006) | Sequence-based | Hybrid GA | Min. Cmax, maximal machine workload and total machine workload | Multiple | FJSP with no-fixed availability constraints. |
| Imanipour (2006) | Sequence-based | Hybrid TS with greedy neighbourhood search | Min. Cmax | Single | FJSP with sequence-dependent setup time |
| Birgin *et al.* (2014) | Sequence-based | - | Min. Cmax | Single | FJSP with sequencing flexibility |
| Gomes* *et al.* (2005) | Time-based | - | Min. Costs derived from failing to meet the 'just in time' due dates, in-process inventory costs and costs of orders not fully completed at the end of the scheduling horizon | Single | FJSP with product re-circulation, parallel homogeneous machines, and intermediate buffers |
| Thomalla (2001) | Time-based | - | Min. Total weighted quadratic tardiness | Single | FJSP in a just-in-time environment |
| Pezzella *et al.* (2008) | - | GA | Min. Cmax | Single | FJSP |
| De Giovanni and Pezzella (2010) | - | GA | Min. Cmax | Single | Distributed and FJSP |

*Table 2.1 Synthesis matrix (continued).*

| Article | Mathematical Model | Algorithm | Objective Function | Single/Multi Objective | Remarks |
|---|---|---|---|---|---|
| Ahmadi *et al.* (2016) | | NRGA | | | FJSP with random machine breakdown |
| Ho *et al.* (2007) | - | LEGA | Min. Cmax | Single | FJSP that integrates evolution and learning within a random search. |
| Demir and İşleyen (2014) | Sequence-based | GA | Min. Cmax | Single | FJSP with overlapping in operations |
| Driss *et al.* (2015) | Sequence-based | New GA | Min. Cmax | Single | FJSP |
| Rossi and Dini (2007) | - | ACO | Min. Cmax | Single | FJSP with routing flexibility, sequence-dependent setup times and transportation time |
| Xing *et al.* (2010) | - | KBACO | Min. Cmax | Single | FJSP |
| Huang *et al.* (2013) | Sequence-based | 2PH-ACO | Min. The sum of weighted earliness and tardiness | Single | FJSP with due window and sequence-dependent setup times |
| Wang *et al.* (2012) | - | ABC | Min. Cmax | Single | FJSP |
| Gao, Peng, *et al.* (2006) | - | GPSO | Min. Cmax | Single | FJSP |
| Zheng and Wang (2016) | Sequence-based | KGFOA | Min. Cmax | Single | FJSP with dual constrained resource |
| Brandimarte (1993) | - | TS | Min. Cmax<br>Min. weighted tardiness | Single | FJSP |
| Hurink *et al.* (1994) | - | TS | Min. Cmax | Single | FJSP with rescheduling |
| Mastrolilli and Gambardella (2000) | - | Two new neighbourhoods functions | Min. Cmax | Single | FJSP |
| Amiri *et al.* (2010) | - | VNS | Min. Cmax | Single | FJSP |
| Yazdani *et al.* (2010) | - | Parallel VNS | Min. Cmax | Single | FJSP |
| Loukil *et al.* (2007) | - | SA | Min. Cmax,<br>Mean completion time, the maximal tardiness, the mean tardiness | Multiple | FJSP with batch production, production of several sub-products and assembly, overlaps of operations |
| Kacem *et al.* (2002b) | - | Pareto approach based on the hybridization of fuzzy logic | Min. Cmax, Workload of the most loaded machine, total workload. | Multiple | FJSP |
| Xie and WuXia and Wu (2005) | - | PSO with SA | Min. Cmax, total workload and critical machine workload | Multiple | FJSP |
| Birgin *et al.* (2015) | - | List scheduling algorithm and a filtered beam search | Min. Cmax | Single | FJSP with sequencing flexibility |
| González *et al.* (2015) | - | Scatter search and path relinking | Min. Cmax | Single | FJSP |
| Li and Gao (2016) | - | Hybridized GA with TS | Min. Cmax | Single | FJSP |

The consideration of sequencing flexibility could give a competitive advantage for industries to decide the sequence that better fits the status of the job shop. To the best of the author's knowledge, only two works have addressed this scenario (Birgin *et al.* 2015; Birgin *et al.* 2014) and developed an MILP formulation and a hybrid metaheuristic respectively.

Dual-resource flexible job shop (DRFJSP) is a variation of the FJSP. This problem considers job sequence, machine assignment, and worker assignment. This scenario is usual for many real-world productions environments; the workers need to perform and control specific tasks on the machines. Skilled workers are typically restricted, due to the high wages and high training cost (Zheng and Wang 2016). These reasons make worker assignment a crucial decision. Research on DRFJSP has been scarce. Only two studies have considered this problem: the single objective work performed by Zheng and Wang (2016) and the multi-objective study provided by Gong *et al.* (2018).

Process plan flexibility is characteristic of modern productions systems. Traditionally, there is a single process plan for a particular job. However, for most manufactured parts, it is possible to generate alternative ways to produce a finished product for obvious reasons. As a result, each feasible way to manufacture a part will create a process plan. Process plan flexibility allows selecting a process plan out of the alternative process plans for a specific job. The selection is implemented to respond to the current product mix and the machine resource availability. Process plan flexibility has not been totally studied for the FJSP. Only three works presented by Özgüven *et al.* (2010), Lee *et al.* (2002) and Kim and Egbelu (1999) dealt with this problem. The authors model this scenario as an MILP formulation. The research on this topic that will be carried out will not consider the process plan generation; it will only consider the possibility of alternative process plans. The selection of the appropriate alternative process plan might be defined by the cost of the process plan. Each process plan cost can vary based on machine tools, fixtures, manufacturing processes, lot sizes, direct and indirect labour cost (Kalpakjian 2001). A selection of the appropriate process plan based on the process plan cost could be incorporated as an objective. At the time of this research, it has been observed that researchers have not been entirely interested in process plan costs for the FJSP that will bring an interesting real-world scenario. Only one study has been found for the classical JSP developed by Brandimarte and Calderini (1995) but not for the FSSP. The research on this topic that will be carried out will not consider the process plan cost calculation, process plan costs will be given, and the process plan that optimizes specific objective will be selected.

Various objective functions have been developed to evaluate the performance of the proposed techniques to deal with the FJSP. The most common objective used is the makespan (Chaudhry and Khan 2016). However, in some cases, the makespan does not reflect the general objective in a manufacturing environment. Usually, companies have to meet due dates. When a due date is not reached, penalties are frequently applied. This situation can generate a huge impact on the different costs. This research direction has been uncommon. Brandimarte (1993) employed the weighted tardiness as the objective to be minimized. Thomalla (2001) minimized the sum of weighted quadratic tardiness. Other objectives as total workload, total tardiness or production cost have been less frequent.

In general, considerations of sequencing flexibility, process plan flexibility, dual-resource approach and processing costs for the FJSP problem have not been fully integrated into previous work.

# CHAPTER 3. FLEXIBLE JOB-SHOP SCHEDULING WITH SEQUENCING FLEXIBILITY (FJSPS)

This chapter defines the FJSP with sequencing flexibility (FJSPS). A biologically inspired algorithm is devised. It is hybridized with a well-known metaheuristics called SA. A proper representation of the problem and the development of novel operators are presented.

## 3.1 Problem description

The FJSP considers $n$ jobs that has to be processed in $m$ machines. Each job consists of a total of $u_j$ operations. Each operation $O_{jk}$ must be both assigned to a machine $i$ and find the operation sequence of the job $j$. Precedence between the operations are given by an arbitrary directed acyclic graph. The objective is to minimize the weighted tardiness.

The following assumptions are proposed for the FJSP:

- All the machines are available at time zero.
- Each machine can perform at most one operation at any time; no multi-tasked is allowed.
- Transportation time is not considered.
- All the jobs are available to schedule at time zero
- Setup time is assumed to be negligible.
- Job preemption is not allowed.

## 3.2 Mathematical formulation

A mixed integer linear programming model is presented to tackle the FJSPS. The notation used for this model is defined as follow:

**Parameters and indexes**

$n$: $Number\ of\ jobs$
$j$: $Index\ of\ jobs\ (1, \dots, n)$
$m$: $Number\ of\ machines$
$i$: $Index\ of\ machines\ (1, \dots, m)$
$u_j$: $Total\ number\ of\ operations\ of\ job\ j$
$O_{jk}$: $O$peration $k$ of job $j$
$k$: $Index\ of\ operations\ (1, \dots, u_j)$
$P_{ijk}$: $Processing\ time\ of\ k$th operation of job $j$ on machine $i$

$e_{jkk'}$: $\begin{cases} 1 \text{ if the } k\text{th operation precedes operation } k' \text{ for job } j \\ 0 \text{ otherwise} \end{cases}$

$d_j$: due date of job $j$

$W_j$: $the\ weight\ of\ job\ j$

$a_{ijk}$: $\begin{cases} 1 \text{ if machine } i \text{ can process the } k\text{th operation of job } j \\ 0 \text{ otherwise} \end{cases}$

**Decision variables**

$x_{ijk}$: $\begin{cases} 1 \text{ if the } k\text{th operation of job } j \text{ is processed on machine } i \\ 0 \text{ otherwise} \end{cases}$

$C_{ijk}$: Completion time $of$ $k$th operation of job $j$ on machine $i$

$T_j$: $tardiness\ of\ job\ j$

$y_{ijkk'}$: $\begin{cases} 1\ if\ operation\ k\ \ precedes\ operation\ k'\ of\ job\ j\ in\ machine\ i \\ 0\ otherwise \end{cases}$

$c_j$: $Completion\ time\ of\ job\ j$

$Q_{ii'jkk'}$: $\begin{cases} 1 \text{ if operation } k \text{ in machine } i \text{ precedes operation } k' \text{ in machine } i'\ of\ job\ j \\ 0 \text{ otherwise} \end{cases}$

$Z_{ijj'kk'}$: $\begin{cases} 1 \text{ if operation } k \text{ of job } j \text{ precedes operation } k' \text{ of job } j' \text{ in machine } i \\ 0 \text{ otherwise} \end{cases}$

The mathematical model is defined as follow:

$$\min \sum_J W_j T_j \tag{3.1}$$

s.t.

$$\sum_{i=1}^{m} x_{ijk} = 1 \qquad \forall\, j, k \tag{3.2}$$

$$C_{i'jk'} \geq C_{ijk} + x_{i'jk'} \cdot P_{i'jk'} - M(1 - x_{i'jk'}) \forall\, i, i', j, k, k' \colon e_{jkk'} = 1, a_{ijk} = 1,\ a_{i'jk'} = 1 \tag{3.3}$$

$$C_{ijk'} \geq C_{ijk} + P_{ijk'} - M(3 - Y_{ijkk'} - x_{ijk} - x_{ijk'}) \ \forall\, i, j, k, k' \colon k \neq k' \tag{3.4}$$

$$C_{ijk} \geq C_{ijk'} + P_{ijk} - M(2 + Y_{ijkk'} - x_{ijk} - x_{ijk'}) \forall\, i, j, k, k' \colon k \neq k' \tag{3.5}$$

$$C_{i'jk'} \geq C_{ijk} + P_{i'jk'} - M(3 - Q_{ii'jkk'} - x_{ijk} - x_{i'jk'}) \forall\, j, k, k' \colon k \neq k', i, i' \colon i \neq i' \tag{3.6}$$

$$C_{ijk} \geq C_{i'jk'} + P_{ijk} - M(2 + Q_{ii'jkk'} - x_{ijk} - x_{i'jk'}) \ \forall\, j, k, k' \colon k \neq k', i, i' \colon i \neq i' \tag{3.7}$$

$$C_{ij'k'} \geq C_{ijk} + P_{ij'k'} - M(3 - Z_{ijj'kk'} - x_{ijk} - x_{ij'k'}) \ \forall\, i, j, j' \colon j \neq j', k, k' \tag{3.8}$$

$$C_{ijk} \geq C_{ij'k'} + P_{ijk} - M(2 + Z_{ijj'kk'} - x_{ijk} - x_{ij'k'}) \ \forall\, i, j, j' \colon j \neq j', k, k' \tag{3.9}$$

$$C_{ijk} - P_{ijk} \geq -M(1 - x_{ijk}) \forall\, i, j, k \tag{3.10}$$

$$C_{ijk} \leq M(x_{ijk}) \forall\, i, j, k \tag{3.11}$$

$$Q_{ii'jkk'} \geq e_{jkk'} \ \ \forall\, j, k, k' \colon k \neq k', i, i' \colon i \neq i' \tag{3.12}$$

$$T_j \geq c_j - d_j \ \forall\, j \tag{3.13}$$

$$T_j \geq 0 \ \forall\, j \tag{3.14}$$

$$c_j \geq C_{ijk} \ \forall\ i, j, k \tag{3.15}$$

$$x_{ijk} \leq a_{ijk} \ \forall\ i, j, k \tag{3.16}$$

$$C_{ijk}, c_j \geq 0 \ \forall\ i, j, k \tag{3.17}$$

$$x_{ijk} \in \{0,1\} \ \forall\ i, j, k \tag{3.18}$$

$$Y_{ijkk'}, Q_{ii'jkk'}, Z_{ijj'kk'} \in \{0,1\} \ \forall\ i, i', j, k, k' \tag{3.19}$$

The objective function is defined by Eq. (3.1) which minimize the weighted tardiness. Constraint set (3.2) ensures that only one machine is selected from the set of available machines to process each operation of a job. Constraint (3.3) states the precedence among the operations of a job. The disjunctive constraint set (Eq. (3.4) and (3.5)) ensures that only one operation can be processed in a machine at a time. The disjunctive constraint set (Eq. (3.6) and (3.7)) avoids the overlapping of operations in different machines for the same job at a time. The disjunctive constraint set (Eq. (3.8) and (3.9)) avoids the overlapping of operations for different jobs for the same machine at a time. Constraint set (Eq. (3.10) and (3.11)) defines the completion time when a machine tool has been assigned. Constraint set (3.12) set specifies the precedence relationship of a job with different machines. Constraint set (3.13) defines the tardiness of a job. Constraint set (3.14) ensures that the tardiness should be either positive or zero. Constraint set (3.15)(3.8) defines the completion time of a job. Constraint set (3.16) specifies the feasibility of the machine assignment. Constraint set (Eq. (3.17), (3.18) and (3.19)) defines the different variable types.

The mathematical model described in Section 3.2 minimizes the weighted tardiness, in case that a new objective, for example, minimize makespan ($C_{max}$) is introduced, Eq. (3.1) is substituted for the following objective function:

$$\min C_{max} \tag{3.20}$$

New constraints are introduced to define makespan ( $C_{max}$) which are described in constraints (3.21) and (3.22):

$$C_{max} \geq c_j \ \forall\ j \tag{3.21}$$

$$C_{max} \geq 0 \tag{3.22}$$

## 3.3 Algorithms

The literature shows that GA has been the most used metaheuristic for the FJSP. However, new metaheuristics are emerging that offers an alternative from traditional methods. For instance, bacterial foraging optimization algorithm is a relative new metaheuristic that is based on the behaviour of a swarm. One process is responsible for the majority of the search, which gives an advantage from other algorithms that employ several processes. In this section, the general algorithm is described.

### 3.3.1 Bacterial foraging optimization algorithm

The Bacterial Foraging Optimization Algorithm (BFOA) is a biologically inspired intelligent random search algorithm which is based on the social foraging behavior of E. Coli bacterium. BFOA was first proposed and developed by Passino (2002). The algorithm emulates the bacteria strategy to seek for food; it tries to maximize the obtained energy per unit of time spent on the foraging process while avoiding noxious substances. The search is done by four processes, namely chemotaxis, swarming, reproduction, and elimination-dispersal which are defined in sub-sections 3.3.1.1-3.3.1.4. The pseudocode for the BFOA can be found in Figure 3.1.

**Begin**
- Initialize $S$, $N_s$, $N_c$, $N_{re}$ and $p_{ed}$
- Execute dispatching rules to generate bacterium $\theta^g(b,f,h)\ \forall\ g,\ \ g = 1...,4$
- Create random initial bacteria $\theta^g(b,f,h)\ \forall\ g,\ \ g = 5...,S$
- Evaluate bacteria.
- **For** $h = 1\ to\ N_{ed}$ **Do**
  - **For** $f = 1\ to\ N_{re}$ **Do**
    - **For** $b = 1\ to\ N_c$ **Do**
      - **For** $g = 1\ to\ S$ **Do**
        - Calculate $C(g)$ and $\phi(g)$
        - *sw* ←1(Counter of swimming length)
        - **While** *sw* ←$N_s$ **Do**
          - Execute chemotaxis process
          - **If** $\theta^g(b+1,f,h)$ is better than $\theta^g(b,f,h)$ **Then**
            - *sw* ← *sw +1*
          - **else**
            - Calculate $C(g)$ and $\phi(g)$
          - **End If**
        - **End Do**
      - **End Do**
    - **End Do**
    - Execute Reproduction process, keeping the best $S/2$ bacteria and create $S/2$ copies of them to maintain a fixed population of bacteria.
  - **End Do**
  - Execute Elimination-Dispersal process with a probability of $0 \leq p_{ed} \leq 1$ in $\theta^g(b,f,h)\ \forall\ g$, $g = 1...,S$.
- **End Do**

**End**

*Figure 3.1 BFOA pseudocode.*

#### 3.3.1.1 Chemotaxis

The displacement of an organism in reaction of a chemical stimulus is called chemotaxis. The Chemotaxis process is performed by a movement named “tumble” which shall find a search direction and followed by a “swim” that allows moving along such direction. The chemotaxis step is defined by Passino (2002) with Eq. (3.23):

$$\theta^g(b+1, f, h) = \theta^g(b, f, h) + C(g)\phi(g) \tag{3.23}$$

where $\theta^g(b, f, h)$ is the location of *g*th bacterium at *b*th chemotaxis step, *f*th reproduction step and, *h*th elimination-dispersal event. $C(g)$ is the step size and $\phi(g)$ its search direction. The total number of bacteria is denoted by $S$. The bacterium will keep swimming in the same direction $N_s$ times only if the new position is better(based on the objective function) than the previous one. However, in case the bacterium position is worst, a new tumble is calculated. The modified chemotaxis for the FJSP is discussed in section 3.5.

#### 3.3.1.2 Swarming

The swarming process is designed to influence the behaviour of the bacteria to explore areas where other population members have found higher nutrients concentrations or to avoid noxious substances. This process is performed using a chemical signalling scheme (attractant and repellant). The process is represented by deforming the search landscape with a set of parameters (Passino 2002).

#### 3.3.1.3 Reproduction

Once maximum chemotaxis steps $N_c$ has been reached, a reproduction event is developed. This process sorts the bacteria swarm based on their objective function value. The healthiest half of the bacteria swarm $S/2$ (the population with the better objective function) will be duplicated and the $S/2$ worst bacteria will be eliminated. This event will maintain a constant population size. The total number of reproductions steps is given by $N_{re}$.

#### 3.3.1.4 Elimination-dispersal

For this event, all the bacteria is carried out to an elimination-dispersal process $N_{ed}$ times with a probability $p_{ed}$.

### 3.3.2 Simulated annealing

Simulated annealing (SA) is a metaheuristic that was first introduced by Kirkpatrick *et al.* (1983). SA has been successfully applied with hybrid metaheuristics in the FJSP (Kaplanoğlu 2016; Shahsavari-Pour and Ghasemishabankareh 2013; Xia and Wu 2005). The SA algorithm can be described as an analogue of an algorithm used in statistical physics for computer simulation on the annealing of a solid to the state with minimum energy (Van Laarhoven *et al.* 1992). The algorithm works by doing small alterations to a current solution. A non-improvement strategy is considered to escape from a local optimum. SA allows accepting the new solution that could be worse than the previous solution with a probability. This procedure is known as the hill climbing method which aims to increase the chances to find the better result of the problem and not being trapped in a local optimum. Figure 3.2 shows a hill climbing representation. One of the control parameters of SA is the *Temperature*. The temperature decreases during the SA execution, and it is responsible for the acceptance probability of worsening solutions.

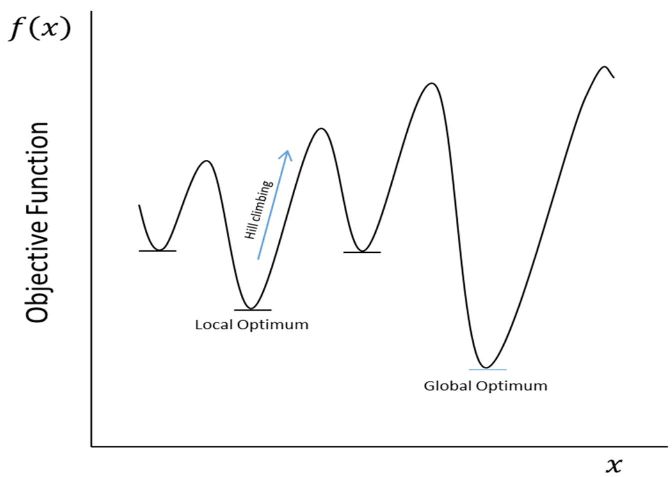



*Figure 3.2 Hill climbing method.*

Starting with an initial solution $A$, SA algorithm generates a new solution $A'$. The new solution is then evaluated and compared with the previous solution by defining $\Delta = f(A') - f(A)$ . For a minimization problem, if $\Delta < 0$, the new solution is accepted. If $\Delta > 0$, then the solution is accepted based on a probability with the function $exp(-\Delta/T)$. Then the new solution is accepted when $rnd < exp(-\Delta/T)$, $rnd$ is a random number in the interval $[0,1]$. $T$ is the temperature parameter, which begins with a high temperature and eventually the temperature is gradually lowered. $T$ is the parameter that controls the acceptance criteria, when $T$ increases, the probability to accept worse solution decreases. $T$ is updated through an exponential function $T =$

$\alpha T$ to slowly reduce the temperature. For each temperature, some iterations are executed which is named epoch length. The algorithm is terminated after a stopping criteria is met. A general SA flowchart can be found in Figure 3.3.

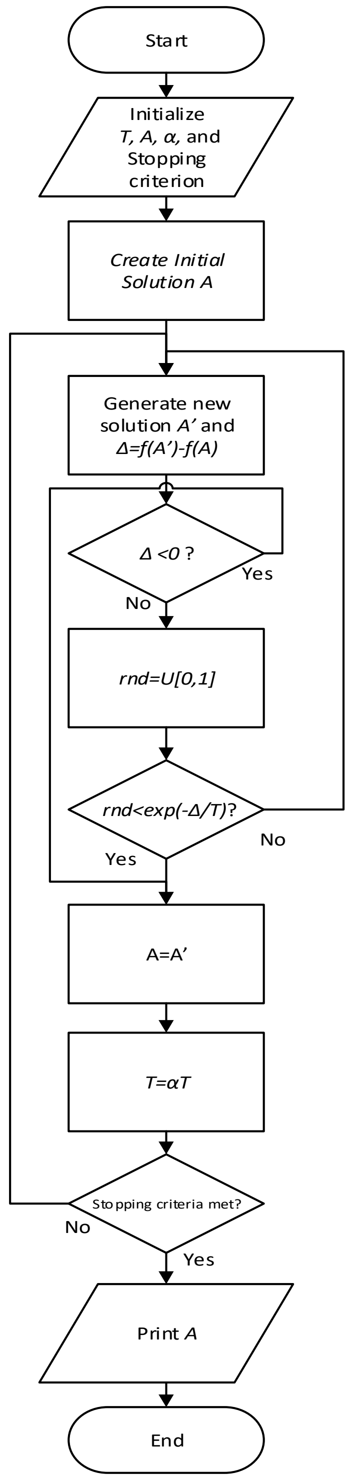



*Figure 3.3 Simulated annealing flowchart.*

## 3.4 Hybrid bacterial foraging optimization algorithm for the FJSP

Initially, the BFOA algorithm has been applied to continuous solution optimization problems. The FJSP is a combinatorial optimization problem, and the BFOA cannot be executed directly. The chemotaxis process needs to be mapped accordingly to a discrete search space as the one for the FJSP. Some attempts have been proposed (Zhao *et al.* 2015; Wu *et al.* 2007) to apply the BFOA in the classical job-shop scheduling problem but the authors do not consider the flexibility of the job-shop. To the best author's knowledge, this is the first application of Hybrid Bacterial Foraging Optimization Algorithm (HBFOA) for FJSP.

The modification to suit the FJSP is done by the development of operators for the chemotaxis and swarming process. In the same manner, SA has been hybridized under the chemotaxis process. Similarly, the reproduction process has modified to avoid premature convergence. The elimination-dispersal process remains as defined on the original BFOA. The pseudocode for the HBFOA for the FJSP is shown in Figure 3.4.

### 3.4.1 Encoding scheme

The following encoding scheme was proposed by Zheng and Wang (2016). The representation of each bacterium $g$ is comprised of two vectors: operation sequence vector (OSV) and machine assignment vector (MAV). The OSV represents the sequence of all operations and all jobs where precedence constrains are satisfied. The MAV denotes the machine assignment for the element (i.e., operation) in the same position on OSV. The length of OSV and MAV is $\sum_j^J u_j$. An example is provided in Table 3.1 and Figure 3.5 to explain the methodology used to encode a feasible solution.

*Table 3.1 Data of instance of 3 jobs and three machines.*

| Job | Operations | Machines $i$ | | | Due Date | Weight |
|---|---|---|---|---|---|---|
| $j$ | $o_{jk}$ | $P_{1jk}$ | $P_{2jk}$ | $P_{3jk}$ | $d_j$ | $W_j$ |
| 1 | $o_{1,1}$ | 7 | 5 | 10 | 5 | 1 |
| | $o_{1,2}$ | 8 | 10 | 5 | | |
| | $o_{1,3}$ | 6 | 5 | 8 | | |
| 2 | $o_{2,1}$ | 8 | 7 | 15 | 6 | 1 |
| | $o_{2,2}$ | 14 | 11 | - | | |
| | $o_{2,3}$ | 5 | 11 | 8 | | |
| | $o_{2,4}$ | 15 | - | 12 | | |
| 3 | $o_{3,1}$ | 6 | 9 | 5 | 8 | 1 |
| | $o_{3,2}$ | 6 | - | 7 | | |
| | $o_{3,3}$ | 10 | - | 9 | | |
| | $o_{3,4}$ | 8 | 7 | 10 | | |

**Begin**

Initialize $S$, $N_{ed}$, $N_s$, $N_c$, $N_{re}$, $p_{re}$ , $p_{ed}$, $T_0$ and $\alpha$

Execute dispatching rules to generate bacterium $\theta^g(b, f, h)\ \forall\ g,\ \ g = 1..., 4$

Create random initial bacteria $\theta^g(b, f, h)\ \forall\ g,\ \ g = 5..., S$ and Evaluate bacteria.

Local Search

$T' \leftarrow T_0$

**For** $h = 1\ to\ N_{ed}$ **Do**

  **For** $f = 1\ to\ N_{re}$ **Do**

    **For** $b = 1\ to\ N_c$ **Do**

      **For** $g = 1\ to\ S$ **Do**

        Calculate $C(g)$ and $\phi(g)$

        *sw* ←1(Counter of swimming length)

        $T \leftarrow T'$

        **While** *sw* ←$N_s$ **Do**

        **If** $probability\ p_{swa}\ criterion\ met$ **then**

          Execute chemotaxis process

          $\Delta \leftarrow f\big(\theta^g(b + 1, f, h)\big) - f\big(\theta^g(b, f, h)\big)$

            **If** $\Delta < 0$ **Then**

              Accept position

            **else**

              **If** $rnd < exp(-\Delta/T)$ **then**

              Accept position

            **else**

              Discard position

            **End if**

              Calculate $C(g)$ and $\phi(g)$

            **End If**

          $T \leftarrow \alpha\mathrm{T}$

          **else**

          Execute Swarming

        **End If**

          *sw* ← *sw +1*

        **End Do**

      **If** $g = S$ **Then**

          $T' \leftarrow T$

      **End if**

      **End Do**

    **End Do**

      Execute Reproduction process with a probability of $0 \leq p_{re} \leq 1$, keeping the best $S/2$ bacteria and create $S/2$ copies of them to maintain a fixed population of bacteria.

    Local Search

  **End Do**

  Execute Elimination-Dispersal process with a probability of $0 \leq p_{ed} \leq 1$ in $\theta^g(b, f, h)\ \forall\ g,\ \ g = 1..., S$.

**End Do**

Local Search

**End**

*Figure 3.4 HBFOA pseudocode.*

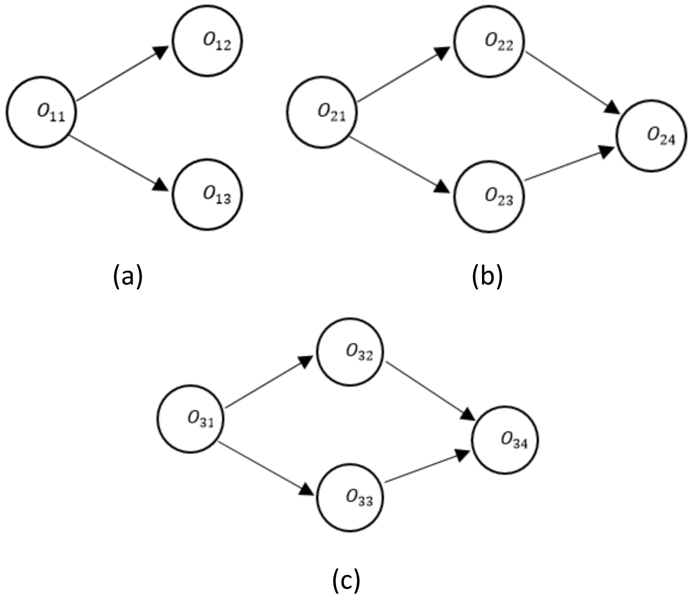



*Figure 3.5 Operation precedence for Table 3.1 (a) Job 1; (b) Job 2 ; (c) Job 3.*

A feasible solution is presented in Figure 3.6, where $o_{1,1}$ is processed first on machine 2, and then $o_{1,2}$ is processed on machine 3 and so on.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{1,1}$ | $o_{1,3}$ | $o_{1,2}$ | $o_{3,1}$ | $o_{3,3}$ | $o_{3,2}$ | $o_{3,4}$ | $o_{2,1}$ | $o_{2,2}$ | $o_{2,3}$ | $o_{2,4}$ |
| MAV | 2 | 2 | 3 | 3 | 3 | 1 | 2 | 2 | 2 | 1 | 3 |

*Figure 3.6 A feasible solution.*

The representation in Figure 3.6 can be decoded with the following procedure:

- Select an operation from OSV one at a time from left to right.
- Schedule the operation to the assigned machine on MAV at earliest as possible.
- This procedure is repeated until all the operations on OSV have been scheduled.

A Gantt chart is presented to illustrate a decoded solution, as per Figure 3.7.

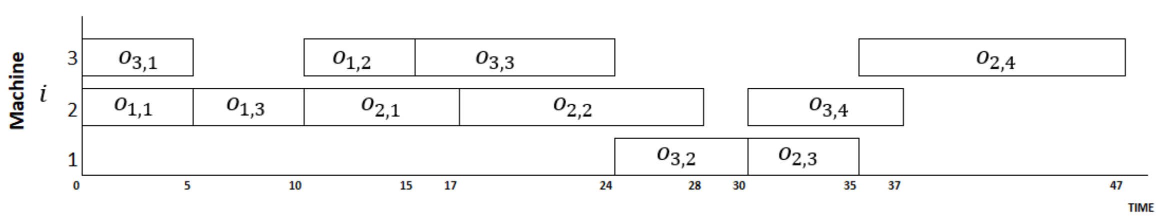



*Figure 3.7 Gantt chart for data of Table 2.*

### 3.4.2 Initial swarm

In order to generate the initial swarm, machine assignment is first produced. The initial assignment for all the swarm is created by randomly selecting three approaches (*AssigmentRule1, AssigmentRule2* and *AssigmentRule3)*. The assignment approaches are described in section 3.4.2. Once that the machine assignments have been decided, the sequencing procedure is performed. Five solutions are sequenced using classical dispatching rules, and the rest of the swarm is obtained by random sequencing. Precedence relationship are enforced in the random sequences generated using a repair mechanism presented in Figure 3.9.

#### 3.4.2.1 Initial machine assignment

Three methods are considered to generate the initial assignment. The first method named *AssigmentRule1,* proposed by Pezzella *et al.* (2008), consists of finding the global minimum in the processing timetable, the assignment is fixed, then a workload update is performed by adding the processing time to the entire column. This workload update is known as the approach by localization (Kacem *et al.* 2002a). The process is repeated until all the operations have been assigned to a capable machine. The second assignment procedure used is *AssigmentRule2* (Pezzella *et al.* 2008) which randomly permute jobs and machines in the processing timetable. Following this order, each operation and each machine is evaluated to find the minimum processing time. The assignment is fixed and then the minimum processing time is added to all the column (machine). The last assignment approach is *AssigmentRule3. AssigmentRule3* corresponds to a slight modification of *AssigmentRule2* where randomly permute jobs, machine and operations. Then the approach by localization (Kacem *et al.* 2002a) is applied. An example is provided in Table 3.2.

*Table 3.2 Initial machine assignment by AssigmentRule3 (machine workload updates in bold).*

| | $P_{2jk}$ | $P_{3jk}$ | $P_{1jk}$ | $P_{2jk}$ | $P_{3jk}$ | $P_{1jk}$ | $P_{2jk}$ | $P_{3jk}$ | $P_{1jk}$ | | $P_{2jk}$ | $P_{3jk}$ | $P_{1jk}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $o_{3,1}$ | 9 | 5 | 6 | 9 | $\underline{5}$ | 6 | 9 | $\underline{5}$ | 6 | | 9 | $\underline{5}$ | 6 |
| $o_{3,2}$ | - | 7 | 6 | - | **12** | 6 | - | 12 | $\underline{6}$ | | - | 7 | $\underline{6}$ |
| $o_{3,4}$ | 7 | 10 | 8 | 7 | **15** | 8 | 7 | 15 | **14** | | $\underline{7}$ | 10 | 8 |
| $o_{3,3}$ | - | 9 | 10 | - | **14** | 10 | - | 14 | **16** | | - | $\underline{9}$ | 10 |
| $o_{2,3}$ | 11 | 8 | 5 | 11 | **13** | 5 | 11 | 13 | **11** | | 11 | 8 | $\underline{5}$ |
| $o_{2,2}$ | 11 | - | 14 | 11 | - | 14 | 11 | - | **20** | … | $\underline{11}$ | - | 14 |
| $o_{2,4}$ | - | 12 | 15 | - | **17** | 15 | - | 17 | **21** | | - | $\underline{12}$ | 15 |
| $o_{2,1}$ | 7 | 15 | 8 | 7 | **20** | 8 | 7 | 20 | **14** | | 7 | 15 | $\underline{8}$ |
| $o_{1,3}$ | 5 | 8 | 6 | 5 | **13** | 6 | 5 | 13 | **12** | | $\underline{5}$ | 8 | 6 |
| $o_{1,1}$ | 5 | 10 | 7 | 5 | **15** | 7 | 5 | 15 | **13** | | 5 | 10 | $\underline{7}$ |
| $o_{1,2}$ | 10 | 5 | 8 | 10 | **10** | 8 | 10 | 10 | **14** | | 10 | $\underline{5}$ | 8 |

### 3.4.2.2 Initial sequencing

Five approaches have been used to generate the initial sequencing. Classical scheduling dispatching rules and a common rule used in the manufacturing sector have been implemented in this work. The rules are described as following:

- **First come first served (FCFS):** Sequences in the same way the jobs arrived.
- **Shortest processing time (SPT):** Sorts the jobs in ascending order based on their total processing time. After this process, the jobs are sequenced following the resulting order.
- **Earliest due date (EDD):** Sorts the jobs in ascending order based on the due date; then the jobs are sequenced following that order. Ties are broken arbitrarily.
- **Minimum number of operations (MNO):** It is a rule used in the manufacturing sector, and it is introduced by the author. This rule is applied when there is a rush for a customer to get their final product. Jobs are sorted in ascending order based on the total number of operations $u_j$, then the jobs are sequenced following that order.
- **Random sequencing (RS):** Sequences the jobs in a random order.

## 3.5 Modified chemotaxis for FJSP

A new scheme is proposed to emulate the chemotaxis event in a combinatorial problem with discrete solution space. When tumbling, the bacteria will find a search direction $\phi(g)$ by randomly selecting one of the following operators: (a) Exchange on the OSV; (b) Exchange on MAV; (c) Exchanges on OSV and MAV. The step size $C(g)$ is defined as a random number of exchanges to perform on the selected operator, with a maximum number of exchanges of $\sum_j^J u_j$ times. In this case, the new position is evaluated and compared with the previous one by calculating $\Delta = f\big(\theta^g(b+1,f,h)\big) - f\big(\theta^g(b,f,h)\big)$. If $\Delta < 0$, the new solution is accepted and a new swim will be performed with the same search direction and step size. However, if $\Delta > 0$ which signifies a neighbor of worse quality, the solution will be accepted only when $rnd < exp(-\Delta/T)$ which is the hill-climbing effect inherited from SA; a new search direction and a step size is computed. The swim is controlled by $N_s$ and the chemotaxis loop is limited by $N_c$ steps. The rest of the SA parameters are computed as follows. The initial temperature $T_0$ is set by Eq. (3.24):

$$T_0 = (N_{ed})(N_{re})(N_c)(N_s)(2) \tag{3.24}$$

The exponential cooling scheme is specified by the following Eq. (3.25):

$$T = \alpha T \tag{3.25}$$

Every bacterium in the swarm has to go through the modified chemotaxis process, in this case, the temperature $T$ has to be reset every time that a new bacterium is processed. For this reason, a new variable $T'$ is introduced to save the value of $T_0$ at the beginning of the HBFOA and $T$ during the swimming process. The algorithm of all the modified chemotaxis can be found in Figure 3.4.

## 3.6 Exchange on the OSV

The exchange is performed by selecting randomly two positions ($r^{th}$ and $s^{th}$) on OSV and then swapping the operation $o_{j,k}$ located at the $r^{th}$ position on OSV to $s^{th}$ position and vice versa. This operator will swap their corresponding MAV as well. This procedure is repeated $C(g)$ times. An example to show an exchange is presented in Figure 3.8.

| | | $r^{th} = 2$ | | | | | | $s^{th} = 8$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{1,1}$ | $\boldsymbol{o_{1,2}}$ | $o_{1,3}$ | $o_{3,1}$ | $o_{3,2}$ | $o_{3,3}$ | $o_{3,4}$ | $\boldsymbol{o_{2,1}}$ | $o_{2,2}$ | $o_{2,3}$ | $o_{2,4}$ |
| MAV | 2 | **3** | 2 | 3 | 1 | 3 | 2 | **2** | 2 | 1 | 3 |

$\Downarrow$

| | | $r^{th} = 2$ | | | | | | $s^{th} = 8$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{1,1}$ | $\boldsymbol{o_{2,1}}$ | $o_{1,3}$ | $o_{3,1}$ | $o_{3,2}$ | $o_{3,3}$ | $o_{3,4}$ | $\boldsymbol{o_{1,2}}$ | $o_{2,2}$ | $o_{2,3}$ | $o_{2,4}$ |
| MAV | 2 | **2** | 2 | 3 | 1 | 3 | 2 | **3** | 2 | 1 | 3 |

*Figure 3.8 Exchange on OSV swapping $r^{th}$=2 and $s^{th}$=8.*

Once the exchange has been done, a repair mechanism is executed to assure that the precedence relationships have not been violated. This mechanism checks the precedence relationship of a pair of operations of the same job; it is engaged in case the prescribed precedence relationship are violated a pairwise exchange is performed. The procedure keeps running until a feasible solution is found. The pseudocode of the repaired mechanism is shown in Figure 3.9.

```
Begin
  repair ← true
  While repair = true Do
    For r=1 to ∑_j^J u_j − 1 Do
      For s=1 to ∑_j^J u_j Do
        If OSV(r) and OSV(s) follow precedence relationship Then
          repair ← false
        else
          repair ← true
          Swap positions rth and sth on OSV and MAV
        End If
      End Do
    End Do
  End Do
End
```

*Figure 3.9 Pseudocode for repair mechanism.*

## 3.7 Exchange on MAV

When the search direction has been set to “Exchange on MAV,” a random position $\boldsymbol{r^{th}}$ on MAV is calculated, and a random machine $i^*$ out of the set of capable machines is computed. The position $\boldsymbol{r^{th}}$ on MAV is updated with $i^*$. This procedure is executed $C(g)$ times.

## 3.8 Exchanges on OSV and MAV

If an exchange on OSV and MAV has been selected as search direction, both procedures explained in previous sections are executed sequentially.

## 3.9 Modified swarming

The discrete solution space structure of the FJSP do not allow to apply the traditional swarming process. The traditional approach aims to explore areas with the highest food concentration. A bacterium located in a high food concentration area signals a “cell-released attractant” to other bacteria. This process allows the bacteria to swarm together. In the same manner, the bacteria signal other bacteria to avoid poor food concentration areas. The process is represented by deforming the search landscape with a set of parameters (Passino 2002). Instead, a modified swarming is presented.

The swarming process is embedded on the chemotaxis swims. This process is executed based on a probability $p_{swa}$, so this process can be done instead of chemotaxis. The probability $p_{swa}$ should be low, because we want to keep using the regular chemotaxis process but in some cases, it is important to guide the bacterium to a most promising area, where some bacteria in the swarm are located. The modified swarming is defined by the crossover of two encoded solutions (original solution and guide solution). The "guide solution" is obtained from a pool where the "best global solution" and the population is stored. A probability is calculated for each solution based on their fitness value. The highest probability will be assigned to the solution with the highest fitness value. Roulette wheel selection is used. This guide solution will be used to produce the swarming solution. Three types of swarming are available, and one case is selected randomly every time that the modified swarming is executed. The swarming types are described below:

- **Swarming OSV:** A set of jobs is chosen randomly, and their positions from the guide solution on the OSV are copied to the swarming solution. The rest of the jobs are copied from left to right from the original solution to the swarming solutions. The machine assignment for all the operations in the swarming solution remains the same as it is established in the original solution. Figure 3.10 shows an example of this swarming.
- **Swarming MAV:** In this case, the original solution is copied to the swarming solution (both OSV and MAV). Then a set of jobs is randomly selected, and the machine assignment of the jobs selected is replaced from the guide solution to the swarming solution on MAV. An example of this approach can be found in Figure 3.11.
- **Total swarming:** A set of jobs is randomly selected, and their positions from the guide solution on the OSV and MAV are copied to the swarming solution. The rest of the jobs are copied from left to right from the original solution to the swarming solutions. This process is shown in Figure 3.12.
-

To avoid copying the whole guide solution to the swarming solution, the maximum number of jobs to be copied is $n-2$.

Original Solution

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{1,1}$ | $o_{1,2}$ | $o_{1,3}$ | $o_{3,1}$ | $o_{3,2}$ | $o_{3,3}$ | $o_{3,4}$ | $o_{2,1}$ | $o_{2,2}$ | $o_{2,3}$ | $o_{2,4}$ |
| MAV | 2 | 3 | 2 | 1 | 3 | 1 | 1 | 2 | 2 | 1 | 3 |

Swarming Solution

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{1,1}$ | $\boldsymbol{o_{3,1}}$ | $o_{1,2}$ | $o_{1,3}$ | $o_{2,1}$ | $\boldsymbol{o_{3,2}}$ | $o_{2,2}$ | $\boldsymbol{o_{3,3}}$ | $o_{2,3}$ | $\boldsymbol{o_{3,4}}$ | $o_{2,4}$ |
| MAV | 2 | **1** | 2 | 3 | 1 | **3** | 2 | **1** | 2 | **1** | 3 |

Guide Solution

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{2,1}$ | $\boldsymbol{o_{3,1}}$ | $o_{1,1}$ | $o_{1,2}$ | $o_{2,2}$ | $\boldsymbol{o_{3,2}}$ | $o_{1,3}$ | $\boldsymbol{o_{3,3}}$ | $o_{2,3}$ | $\boldsymbol{o_{3,4}}$ | $o_{2,4}$ |
| MAV | 2 | **3** | 1 | 3 | 2 | **1** | 2 | **3** | 1 | **2** | 3 |

*Figure 3.10 Swarming OSV ( j=3).*

Original Solution

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{1,1}$ | $o_{1,2}$ | $o_{1,3}$ | $o_{3,1}$ | $o_{3,2}$ | $o_{3,3}$ | $o_{3,4}$ | $o_{2,1}$ | $o_{2,2}$ | $o_{2,3}$ | $o_{2,4}$ |
| MAV | 2 | 3 | 2 | 1 | 3 | 1 | 1 | 2 | 2 | 1 | 3 |

Swarming Solution

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{1,1}$ | $o_{1,2}$ | $o_{1,3}$ | $o_{3,1}$ | $o_{3,2}$ | $o_{3,3}$ | $o_{3,4}$ | $o_{2,1}$ | $o_{2,2}$ | $o_{2,3}$ | $o_{2,4}$ |
| MAV | 2 | 3 | 2 | **3** | **1** | **3** | **2** | 2 | 2 | 1 | 3 |

Guide Solution

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{2,1}$ | $\boldsymbol{o_{3,1}}$ | $o_{1,1}$ | $o_{1,2}$ | $o_{2,2}$ | $\boldsymbol{o_{3,2}}$ | $o_{1,3}$ | $\boldsymbol{o_{3,3}}$ | $o_{2,3}$ | $\boldsymbol{o_{3,4}}$ | $o_{2,4}$ |
| MAV | 2 | **3** | 1 | 3 | 2 | **1** | 2 | **3** | 1 | **2** | 3 |

*Figure 3.11 Swarming MAV (j=3).*

Original Solution

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{1,1}$ | $o_{1,2}$ | $o_{1,3}$ | $o_{3,1}$ | $o_{3,2}$ | $o_{3,3}$ | $o_{3,4}$ | $o_{2,1}$ | $o_{2,2}$ | $o_{2,3}$ | $o_{2,4}$ |
| MAV | 2 | 3 | 2 | 3 | 1 | 3 | 2 | 2 | 2 | 1 | 3 |

Swarming Solution

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{1,1}$ | $\boldsymbol{o_{3,1}}$ | $o_{1,2}$ | $o_{1,3}$ | $o_{2,1}$ | $\boldsymbol{o_{3,2}}$ | $o_{2,2}$ | $\boldsymbol{o_{3,3}}$ | $o_{2,3}$ | $\boldsymbol{o_{3,4}}$ | $o_{2,4}$ |
| MAV | 2 | **3** | 2 | 3 | 1 | **1** | 2 | **3** | 2 | **2** | 3 |

Guide Solution

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{2,1}$ | $\boldsymbol{o_{3,1}}$ | $o_{1,1}$ | $o_{1,2}$ | $o_{2,2}$ | $\boldsymbol{o_{3,2}}$ | $o_{1,3}$ | $\boldsymbol{o_{3,3}}$ | $o_{2,3}$ | $\boldsymbol{o_{3,4}}$ | $o_{2,4}$ |
| MAV | 2 | **3** | 1 | 3 | 2 | **1** | 2 | **3** | 1 | **2** | 3 |

*Figure 3.12 Total Swarming (j=3).*

## 3.10 Modified reproduction

The traditional reproduction process limits the swarm diversity. The bacteria duplication could lead to premature convergence (Hernández-Ocana *et al.* 2016). To overcome this situation, the reproduction process will be executed based on a probability $p_{re}$.

## 3.11 Critical path theory

A feasible solution of the FJSP can be defined using a disjunctive graph $G = (N, A, F)$ where $N$ represents the set of operations including the dummy nodes $S$ and $T$ for starting and terminations respectively. The set $A$ is comprised by conjunctive arcs (real arcs) which symbolises the imposed precedence relationships (technological precendence constrainsts) and the, set $F$ contains disjunctive arcs (dashed arcs) that represents operations executed on the same machine. The processing time $P_{ijk}$ is shown on each arc. For example, Figure 3.13 shows a schedule of a feasible solution of Figure 3.6.

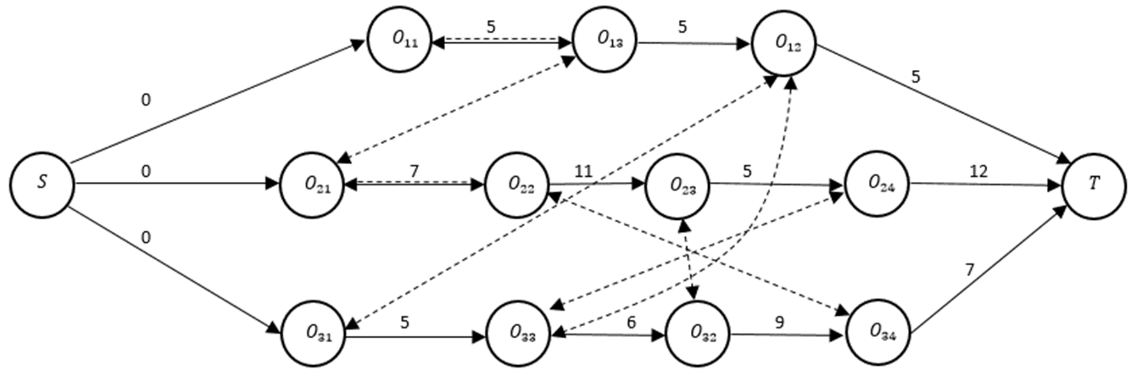



Figure 3.13 Illustration of a disjunctive graph.

Once the directions of the disjunctive arcs have been chosen which was brought by the sequencing result on each machine, a directed acyclic graph (DAG) is obtained. Figure 3.14 shows an example of a DAG.

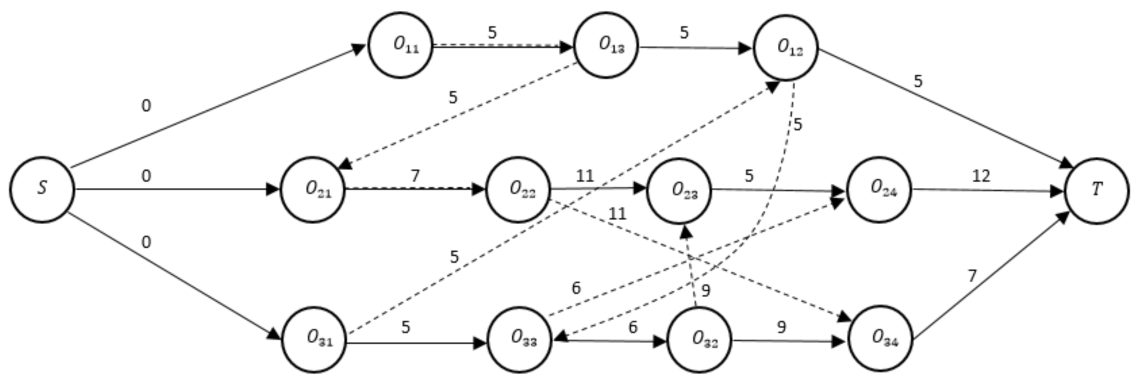



Figure 3.14 Illustration of a DAG.

The critical path is defined as the longest path in the disjunctive graph. Hence the makespan is no shorter than any possible path in the DAG. The operations on the critical path are named critical operations. A critical path cannot be delayed without increasing the makespan of the schedule (Gen *et al.* 2009). The critical path is presented in with bold arcs and highlighted nodes in Figure 3.15 for the feasible solution in Figure 3.6. The single-source shortest path algorithm could be used to find the critical path. The algorithm will be briefly explained in the following sections.

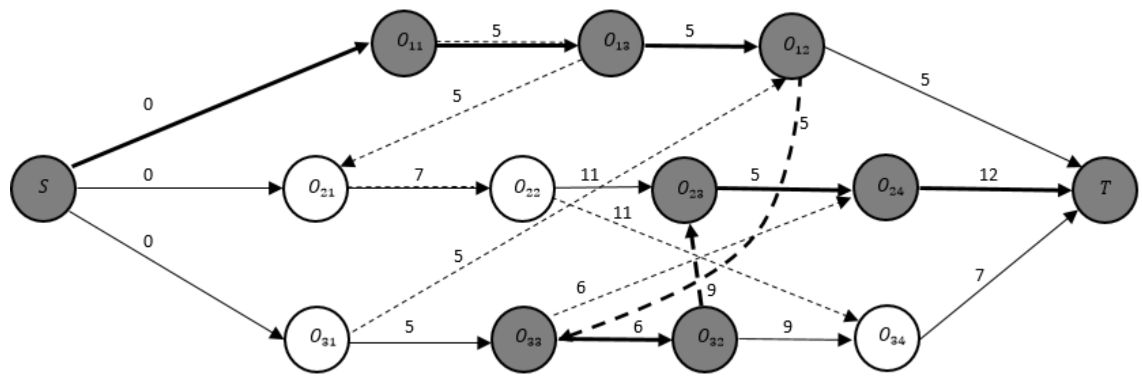



*Figure 3.15 Illustration of the critical path.*

### 3.11.1 Topological sort

A topological sort is a crucial procedure to calculate the critical path. For simplicity, it is assumed that $E = A \cup F.$ A topological sort of a DAG $G = (N, E)$ is a linear ordering of all its arcs such that if $G$ contains an arc $(u, v)$, the $u$ appears before $v$ in the ordering (Cormen *et al.* 2001). This type of sort can be observed as a feasible solution like the one explained in Figure 3.6, that is a sequential order of all the nodes (operations) in a way to satisfy the precedence constraints. It is important to calculate different topological of DAG to be able to find alternative critical paths.

The algorithm to get a topological sort of a DAG starts by marking all the nodes as not visited, then a node $v\, in\, N$ is randomly selected. If $v$ is not visited, a recursive function *TopologicalSortUntil* is called. The node is marked as visited and all its adjacent nodes not visited are explored with the *TopologicalSortUntil*. Once that all adjacent nodes are visited, the current node is added to the $Stack$. The $Stack$ will contain all the nodes in a topological sort. Figure 3.16 and Figure 3.17 shows the topological sort algorithm and the function *TopologicalSortUntil* respectively.

```
Begin
   N ← Set of nodes
   Adj() ← Set of adjacent vertices
   Visited() ← Boolean array to define when a  vertex is visited
   For each v in N Do
     visited (v) ← False
   End Do
   For each v in N Do
     If visited(v) = False then
       TopologicalSortUntil(v)
     End If
   End Do
End
```

*Figure 3.16 Topological sort algorithm.*

```
Begin TopolgicalSortUntil(u)
  visited (u) ← True
  For each  i in  Adj (u) Do
    If visited(i) = False then
      TopologicalSortUntil(i)
    End If
    Insert u in Stack
  End Do
End
```

*Figure 3.17 TopologicalSortUntil Function.*

### 3.11.2 Single-source shortest path in a DAG

The longest path through the DAG is known as the critical path. Single-source shortest path can be used to find the critical path by negating the weight (processing time) of each arc. After applying Single-source shortest path, the critical path can be retrieved by the predecessor list $\pi(u)$. The algorithm explores each vertex in the topologically sorted order and the distance at each node is updated. The pseudocode for a single-source shortest path in a DAG can be found in Figure 3.18.

```
Begin
  N ← Set of nodes
  dist(u) ← Distance at vertex u
  Adj() ← Set of adjacent vertices
  weight(u,v) ← Weight from node u to v
  s ← source node
  π(u) ← predecessor of vertex u
  For each v in N Do
    dist(v) ← ∞
  End Do
  dist(s) ← 0
  For each u in N , taken in a topologically sorted order Do
    For each  v in  Adj (u) Do
      If dist(v) > dist(u) + weight(u,v) then
        dist(v) ← dist(u) + weight(u,v)
        π(v) ← u
      End If
    End Do
  End Do
End
```

*Figure 3.18 Single-source shortest path in DAG.*

## 3.12 Local search

Local search techniques have been incorporated into different metaheuristics to improve the solution quality and to reduce the computational time. For any local search technique, the definition of an effective neighbourhood structure is required to implement it. Five neighbourhood functions have been developed. The neighbourhood functions are described in the following section. The neighbourhood functions are embedded in the HBFOA, and they are deployed systematically during the different stages in the metaheuristic. The local search method is applied after the initial swarm is created. Only the best, worst and the $S/2$th bacteria are improved by local search with a maximum number of iteration $LS_1$ . The same approach is applied before finishing the reproduction loop with a maximum number of iteration $LS_2$. At the completion of the HBFOA, the three best solutions are further improved using the local search method with a maximum number of iteration $LS_1$.

$As_1$, $As_3$, $Se_1$and $Se_2$ are employed for the case of the minimization of the weighted tardiness and $As_1$, $As_2$, $As_3$, $Se_1$and $Se_2$ are used for the minimization of makespan.

The use of the neighbourhood functions is sequentially activated (either machine assignment or operation sequence function). The neighbourhood functions are switched after five consecutive non-improvement. The local search procedure is finished until a maximum number of iterations ($MaxIter$) is achieved. The pseudocode of the local search procedure is presented in Figure 3.19.

```
Begin
  SeqSearch ← True
  While Iter< MaxIter Do
    If  SeqSearch = true  then
      Randomly deploy a sequence neighbourhood function
    Else
      Randomly deploy a machine assignment neighbourhood function
    End If
    If  SeqSearch = true  then
      SeqSearch ← False
    Else
      SeqSearch ← True
    End If
  End Do
End
```

*Figure 3.19 Local search pseudocode.*

### 3.12.1 Neighbourhood function $As_1$

The neighbourhood function $As_1$ is based on the concept of the machine workload and our efforts to balance the total workload in all the machines. A critical operation is randomly selected from the machine with the highest workload and is assigned to that with the lowest workload, if possible. If the machine with the lowest workload is not capable to process the selected operation, the operation is assigned sequentially based on the workload.

### 3.12.2 Neighbourhood function $As_2$

The neighbourhood function $As_2$ is created by selecting a random critical operation and then it is assigned to an alternative machine out of a set of capable machines, if possible.

### 3.12.3 Neighbourhood function $As_3$

The neighbourhood function $As_3$ is executed by choosing a random operation and then is randomly assigned to another capable machine.

### 3.12.4 Neighbourhood function $Se_1$

For this neighbourhood function, a random critical operation is selected and is swapped with its adjacent operation on the same machine. The swap must satisfy the precedence constraints.

### 3.12.5 Neighbourhood function $Se_2$

This neighbourhood function swaps consecutive operations in the same machine while preserving precedence relationships. The operations are randomly selected.

## 3.13 Summary

Flexible sequencing has been integrated into the FJSP. It emphasizes that precedence relationships are not always in sequential order. The precedence relationships can be given by a DAG. A single objective mathematical formulation has been devised where the minimization of the makespan and weighted tardiness has been studied. A HBFOA has been proposed to tackle the FJSPS. The algorithm integrates the SA algorithm. In addition, a new rule-based procedure called Minimum number of operations has been developed as part of the initial assignment. Novels neighbourhood function based on the critical path are presented as a local search procedure. Experimentation is developed in CHAPTER 7.

# CHAPTER 4. DUAL-RESOURCE FLEXIBLE JOB-SHOP SCHEDULING PROBLEM WITH SEQUENCING FLEXIBILITY (DRFJSPS)

In practical production, a machine is usually operated by a worker. Each worker has the ability to operate a particular set of machines. Figure 4.1 illustrates this common scenario. For example, worker 2 is capable of operating machines 2 and 3. In the same manner, operation 2 can be processed on machine 2 and 3. This feature is known as worker flexibility (Gong *et al.* 2018). Skilled workers are typically limited because of high wages and absence of trained labour (Zheng and Wang 2016). All these circumstances lead the manager with a complex problem in addition to the intractable FJSPS.

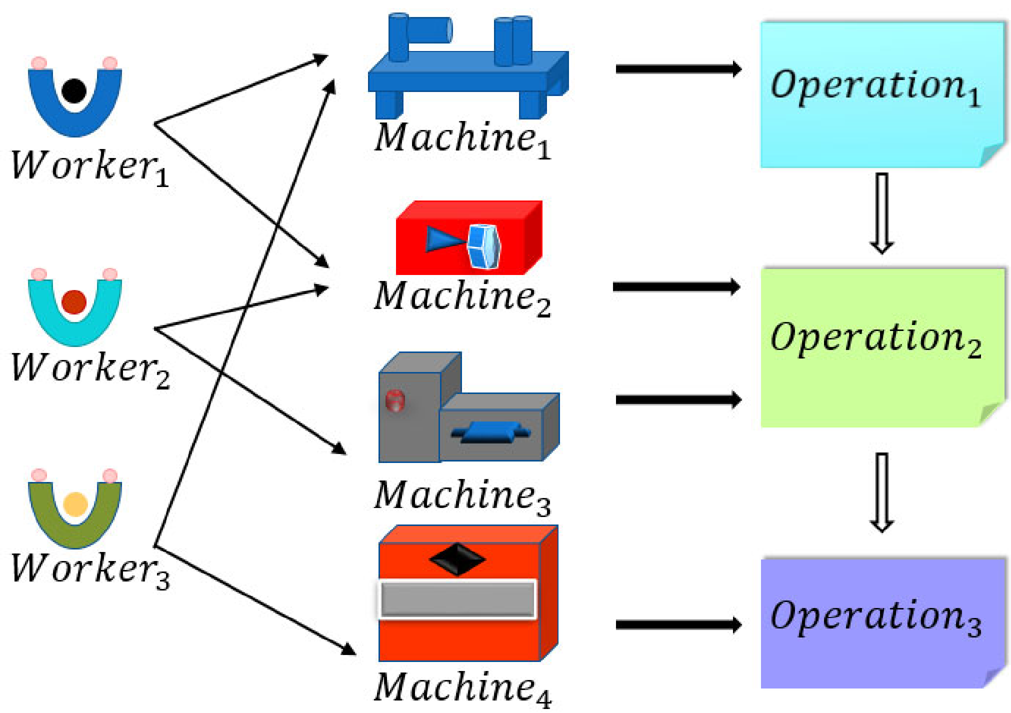



*Figure 4.1 Practical the dual job-shop scheduling model.*

The decision about machine and worker assignment is crucial, and it enormously impacts the production efficiency. Managers have to come with a best resources allocation and operation sequencing based on the optimization objectives. There is a need to deliver the jobs as fast as possible while using the job-shop resources efficiently. The makespan gives a proper measure to deal with this problem. Similarly, it is essential to balance the workload of the worker. It cannot be possible to find workers with the very high workload and at the same time workers with a low workload. In this case, we can avoid this problem by minimizing the most loaded worker. Regularly, companies have due dates to accomplish. In case they cannot meet this requirement, a penalty is issued based on the number of tardy days. Weighted tardiness can be used as an indicator to avoid this situation. A multi-objective approach could capture the mentioned needs for this problem.

## 4.1 Problem description

The Dual-resource flexible job-shop scheduling problem with sequencing flexibility (DRFJSPS) with sequencing flexibility considers $n$ jobs that have to be processed in $m$ machines by set of workers $U$. Each job consists of a total of $u_j$ operations. Each operation $O_{jk}$ must be both assigned to a machine $i$ and a worker $u$. Precedence between the operations are given by an arbitrary directed acyclic graph. The objectives in the multi-objective DRFJSPS with sequencing flexibility are the minimization of the following criteria:

- Makespan
- Maximal worker workload
- Weighted tardiness

The following assumptions are proposed for the FJSP:

- All the machines and workers are available at time zero.
- The worker may be transferred from one machine to another.
- Each machine can process only one operation at a time on any job.
- Each worker can operate only one machine at a time for any operation.
- Each operation can be performed only once on one machine, and its sequence is respected for every job.
- Transportation time is not considered.
- The operations of different jobs do not have precedence constraints.
- Job preemption is not allowed.
- An operation of any job cannot be processed until its preceding operations are completed.
- The processing time corresponding to the jobs, operations, machines, and workers are given in advance.

## 4.2 Mathematical formulation

A mixed integer linear programming model is presented to tackle the DRFJSPS. The notation required for the model is given as follows:

**Parameters and indexes**

$n$: Number of jobs
$j$: Index of jobs $(1, \ldots, n)$
$m$: Number of machines
$i$: Index of machines $(1, \ldots, m)$
$U$: Number of jobs
$u$: Index of workers $(1, \ldots, U)$
$u_j$: Total number of operations of job $j$
$O_{jk}$: $O$peration $k$ of job $j$
$k$: Index of operations $(1, \ldots, u_j)$

$P_{iujk}$: Processing time of operation $k$ of job $j$ machine $i$ by worker $u$
$e_{jkk'}$: $\begin{cases} 1 \text{ if operation } k \text{ precedes operation } k' \text{ for job } j \\ 0 \quad \text{otherwise} \end{cases}$
$d_j$: due date of job $j$
$W_j$: the weight of job $j$
$a_{iujk}$: $\begin{cases} 1 \text{ if machine } i \text{ can process the operation } k \text{ of job } j \text{ by worker } u \\ 0 \quad \text{otherwise} \end{cases}$

**Decision variables**

$x_{iujk}$: $\begin{cases} 1 \text{ if the } k\text{th operation of job } j \text{ is processed on machine } i \text{ by worker } u \\ 0 \quad \text{otherwise} \end{cases}$
$C_{iujk}$: Completion time $of$ $k$th operation of job $j$ on machine $i$ by operator $u$
$T_j$: tardiness of job $j$
$c_j$: Completion time of job $j$
$Y_{iujkk'}$: $\begin{cases} 1 \text{ if the } k\text{th operation precedes operation } k' \text{ of job } j \text{ processed on machine } i \\ \qquad \text{by worker } u \\ 0 \quad \text{otherwise} \end{cases}$
$Q_{ii'uu'jkk'}$: $\begin{cases} 1 \text{ if operation } k \text{ processed on machine } i \text{ by worker } u \text{ precedes operation } k' \\ \qquad \text{processed on machine } i' \text{ by worker } u' \text{ of job } j \\ 0 \quad \text{otherwise} \end{cases}$
$Z_{iuu'jj'kk'}$: $\begin{cases} 1 \text{ if operation } k \text{ of job } j \text{ processed by worker } u \text{ precedes operation } k' \text{ of job } j' \\ \qquad \text{processed by worker } u' \text{ on machine } i \\ 0 \quad \text{otherwise} \end{cases}$
$R_{ii'ujj'kk'}$: $\begin{cases} 1 \text{ if operation } k \text{ of job } j \text{ processed on machine } i \text{ precedes operation } k' \text{ of job } j' \\ \qquad \text{processed on machine } i' \text{ by worker } u \\ 0 \quad \text{otherwise} \end{cases}$
$S_{iuu'jkk'}$: $\begin{cases} 1 \text{ if operation } k \text{ processed by worker } u \text{ precedes operation } k' \text{ processed by} \\ \qquad \text{worker } u' \text{ on machine } i \text{ of job } j \\ 0 \quad \text{otherwise} \end{cases}$
$A_{iujj'kk'}$: $\begin{cases} 1 \text{ if operation } k \text{ of job } j \text{ precedes operation } k' \text{ of job } j' \text{ processed on machine } i \\ \qquad \text{by worker } u \\ 0 \quad \text{otherwise} \end{cases}$

$$B_{ii'ujkk'}:\begin{cases}1 \text{ if operation } k \text{ processed on machine } i \text{ precedes operation } k' \text{ processed} \\ \qquad\qquad \text{on machine } i' \text{ by worker } u \text{ of job } j \\ 0 \quad \text{otherwise}\end{cases}$$

The mathematical model is defined as follows:

$$min\ f_1 = c_{max} \tag{4.1}$$

$$min\ f_2 = L_{max} \tag{4.2}$$

$$min\ f_3 = \sum_{j=1}^{n} W_j T_j \tag{4.3}$$

s.t.

$$\sum_{u=1}^{U} \sum_{i=1}^{m} x_{iujk} = 1 \qquad \forall\, j, k \tag{4.4}$$

$$C_{i'u'jk'} \geq C_{iujk} + x_{i'u'jk'} \cdot P_{i'u'jk'} - M\left(1 - x_{i'u'jk'}\right) \forall\, i, i', u, u', j, k \neq k', e_{jkk'} = 1, a_{iujk} = 1, \quad a_{i'u'jk'} = 1 \tag{4.5}$$

$$C_{iujk'} \geq C_{iujk} + P_{iujk'} - M(3 - Y_{iujkk'} - x_{iujk} - x_{iujk'})\ \forall\, i, u, j, k \neq k' \tag{4.6}$$

$$C_{iujk} \geq C_{iujk'} + P_{iujk} - M\left(2 + Y_{iujkk'} - x_{iujk} - x_{iujk'}\right) \forall\, i, j, k \neq k' \tag{4.7}$$

$$C_{i'u'jk'} \geq C_{iujk} + P_{i'u'jk'} - M\left(3 - Q_{ii'uu'jkk'} - x_{iujk} - x_{i'u'jk'}\right) \forall\, i \neq i', u \neq u', j, k \neq k' \tag{4.8}$$

$$C_{iujk} \geq C_{i'u'jk'} + P_{iujk} - M\left(2 + Q_{ii'uu'jkk'} - x_{iujk} - x_{i'u'jk'}\right) \forall\ i \neq i', u \neq u', j, k \neq k' \tag{4.9}$$

$$C_{iu'j'k'} \geq C_{iujk} + P_{iu'j'k'} - M\left(3 - Z_{iuu'jj'kk'} - x_{iujk} - x_{iu'j'k'}\right) \forall\, i, u \neq u', j \neq j', k \tag{4.10}$$

$$C_{iujk} \geq C_{iu'j'k'} + P_{iujk} - M\left(2 + Z_{iuu'jj'kk'} - x_{iujk} - x_{iu'j'k'}\right) \forall\, i, u \neq u', j \neq j', k \tag{4.11}$$

$$C_{i'uj'k'} \geq C_{iujk} + P_{i'uj'k'} - M\left(3 - R_{ii'ujj'kk'} - x_{iujk} - x_{i'uj'k'}\right) \forall\, i \neq i', u, j \neq j', k \tag{4.12}$$

$$C_{iujk} \geq C_{i'uj'k'} + P_{iujk} - M\left(2 + R_{ii'ujj'kk'} - x_{iujk} - x_{i'uj'k'}\right) \forall\, i \neq i', u, j \neq j', k \tag{4.13}$$

$$C_{iu'jk'} \geq C_{iujk} + P_{iu'jk'} - M\left(3 - S_{iuu'jkk'} - x_{iujk} - x_{iu'jk'}\right) \forall\, i, u \neq u', j, k \neq k' \tag{4.14}$$

$$C_{iujk} \geq C_{iu'jk'} + P_{iujk} - M\left(2 + S_{iuu'jkk'} - x_{iujk} - x_{iu'jk'}\right) \forall\, i, u \neq u', j, k \neq k' \tag{4.15}$$

$$C_{iuj'k'} \geq C_{iujk} + P_{iuj'k'} - M\left(3 - A_{iujj'kk'} - x_{iujk} - x_{uj'k'}\right) \forall\, i, u, j \neq j', k \tag{4.16}$$

$$C_{iujk} \geq C_{iuj'k'} + P_{iujk} - M\left(2 + A_{iujj'kk'} - x_{iujk} - x_{uj'k'}\right) \forall\, i, u, j \neq j', k \tag{4.17}$$

$$C_{i'ujk'} \geq C_{iujk} + P_{i'ujk'} - M\left(3 - B_{ii'ujkk'} - x_{iujk} - x_{i'ujk'}\right) \forall\ i \neq i', u, j, k \tag{4.18}$$

$$C_{iujk} \geq C_{i'ujk'} + P_{i'ujk'} - M\left(2 + B_{ii'ujkk'} - x_{iujk} - x_{i'ujk'}\right) \forall\ i \neq i', u, j, k \tag{4.19}$$

$$x_{iujk} \leq a_{iujk} \; \forall \; i, u, j, k \tag{4.20}$$

$$C_{iujk} - P_{iujk} \geq \; -M\left(1 - x_{iujk}\right) \forall \; i, u, j, k \tag{4.21}$$

$$C_{iujk} \leq M\left(x_{iujk}\right) \forall \; i, u, j, k \tag{4.22}$$

$$c_{max} \geq C_{iujk} \; \forall \; i, u, j, k \tag{4.23}$$

$$L_u = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k}^{u_j} x_{iujk} P_{iujk} \forall \; u \tag{4.24}$$

$$L_{max} \geq L_u \forall \; u \tag{4.25}$$

$$T_j \geq c_j - d_j \; \forall \; j \tag{4.26}$$

$$T_j \geq 0 \; \forall \tag{4.27}$$

$$c_j \geq C_{iujk} \; \forall \; i, u, j, k \tag{4.28}$$

$$C_{iujk} \geq 0 \forall \; i, j, u, k \tag{4.29}$$

$$x_{iujk}, Y_{iujkk'}, Q_{ii'uu'jkk'}, Z_{iuu'jj'kk'}, R_{ii'ujj'kk'}, S_{iuu'jkk'}, A_{iujj'kk'}, B_{ii'ujkk'} \in \{0,1\} \; \forall \; i, i', u, \quad u', j, j'k, k' \tag{4.30}$$

The first objective is defined by Eq. (4.1) which minimizes the makespan. The second objective is presented in Eq. (4.2) that minimizes the maximal worker workload. The third objective is shown in Eq. (4.3) that minimizes the weighted tardiness. Constraint set (4.4) guarantees that only one machine and only one worker are selected to process each operation of a job. Constraint set (4.5) defines the precedence among the operations of a job. Constraint set (4.7) ensures that only one operation can be processed on a machine by a worker at a time. Constraint set (Eq. (4.8) and (4.9)) avoids the overlapping of operations in different machines and workers for the same job at a time. The disjunctive constraint set (Eq. (4.10) and (4.11)) avoids the overlapping of operations for different jobs and workers for the same machine at a time. The disjunctive constraint set (Eq. (4.12) and (4.13)) avoids the overlapping of operations in different machines and jobs processed by the same operator at a time. Constraint set (Eq. (4.14) and (4.15)) prevents overlapping of operations processed by different workers for the job and machine at a time. Constraint set (Eq. (4.16) and (4.17)) avoids overlapping of operations of different jobs in the same machine and by the same worker. Constraint set (Eq. (4.18) and (4.19)) prevents overlapping of operations for the same job in different machines processed by the same worker. Constraint set (4.20) defines the feasibility of the machine assignment. Constraint set (Eq. (4.21) and (4.22)) defines the completion time when a machine tool has been assigned. Constraint set (4.23) states the makespan. Constraint set (4.24) defines the workload of a worker. Constraint set (4.25) defines the most loaded worker. Constraint set (Eq. (4.26) and (4.27)) specifies the tardiness of a job. Constraint set (4.28) defines the completion time of a job. Constraint set (Eq. (4.29) and (4.30)) defines the different variable types.

## 4.3 Illustrative example

In order to integrate the multiple objectives, the weighted sum approach has been used, and it can be found in Eq. (4.31). The weights $w_1'$, $w_2'$ and $w_3'$ are input data provided by the user.

$$min\ f' = w_1' f_1 + w_2' f_2 + w_3' f_3 \tag{4.31}$$

Eq. (4.31) employs three weights ($w_1'$, $w_2'$ and $w_3'$).

In Table 4.1, an illustrative instance is presented. It shows the processing times for the operations for DRFJSPS with three jobs, three machines, and three workers. It can be observed that the processing time varies depending on the eligible machine and worker. The symbol "-" is used to refer that the operation cannot be executed by the eligible machine and worker. For example, operation $o_{3,1}$ can be processed on machine 1 ($i = 1$) by worker 2 ($u = 2$) but it cannot be processed on machine 1 ($i = 1$) by worker 2 ($u = 3$); similarly operation $o_{2,1}$ can be processed on machine 2 ($i = 2$) by worker 1 ($u = 1$) and worker 2 ($u = 2$). Precedence relationships for this illustrative example can be found in Figure 4.2.

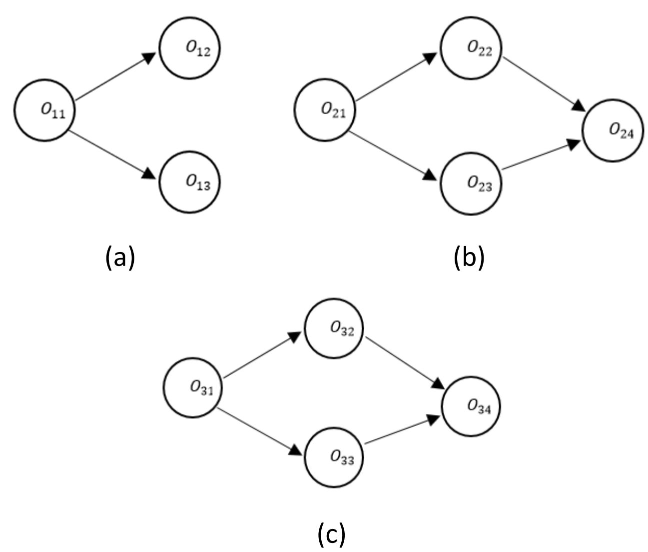



*Figure 4.2 Operation precedence for Table 4.1 (a) Job 1; (b) Job 2 ; (c) Job 3.*

*Table 4.1 Data of an illustrative instance.*

| Job | Operations | Machine $i = 1$ | | | Machine $i = 2$ | | | Machine $i = 3$ | | | Due Date | Weight |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $j$ | $o_{jk}$ | $P_{11jk}$ | $P_{12jk}$ | $P_{13jk}$ | $P_{21jk}$ | $P_{22jk}$ | $P_{23jk}$ | $P_{31jk}$ | $P_{32jk}$ | $P_{33jk}$ | $d_j$ | $W_j$ |
| 1 | $o_{1,1}$ | - | 11 | - | 10 | - | 5 | 14 | - | 5 | 6 | 1 |
| | $o_{1,2}$ | 8 | - | 6 | 9 | - | 14 | - | 8 | 15 | | |
| | $o_{1,3}$ | 10 | - | - | 11 | 11 | - | - | - | - | | |
| 2 | $o_{2,1}$ | - | 15 | - | 15 | 7 | - | - | - | 5 | 5 | 1 |
| | $o_{2,2}$ | 11 | - | - | 7 | - | 10 | - | - | 14 | | |
| | $o_{2,3}$ | - | 14 | - | 8 | 6 | 6 | - | - | 8 | | |
| | $o_{2,4}$ | 9 | 13 | 14 | - | - | 12 | 8 | - | 13 | | |
| 3 | $o_{3,1}$ | - | 15 | - | 10 | - | - | - | - | - | 8 | 1 |
| | $o_{3,2}$ | - | - | 8 | - | 9 | - | - | 8 | 14 | | |
| | $o_{3,3}$ | 8 | 10 | - | 13 | - | - | - | - | 14 | | |
| | $o_{3,4}$ | - | 5 | 6 | - | 10 | - | - | 11 | - | | |

Xpress Optimizer 25.01.05 algebraic model language and optimizer have been used to code the mathematical model. The weights in Eq. (4.31) have been set to $w_1' = 0.3$, $w_1' = 0.3$ and $w_1' = 0.4$. The optimal solution has been found after 246 seconds. The value of different objectives is presented in Table 4.2. Two Gantt charts have been developed. The first one is shown in Figure 4.3 and plots time against machines. The second one can be seen in Figure 4.4 and plots time against worker.

*Table 4.2 Objective values for the illustrative example.*

| Objective | | Value |
|---|---|---|
| Makespan | $f_1 = c_{max}$ | 30 |
| Maximal worker workload | $f_2 = L_{max}$ | 28 |
| Weighted tardiness | $f_3 = \sum_{j=1}^{n} W_j T_j$ | 63 |

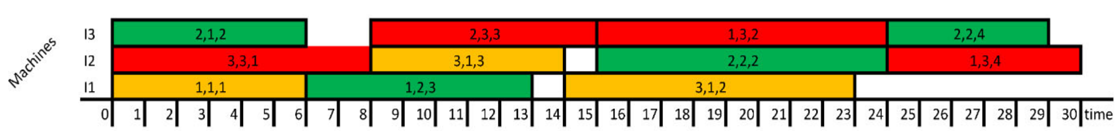



*Figure 4.3 Machine Gantt chart for illustrative example. Labels per operation (u,j,k).*

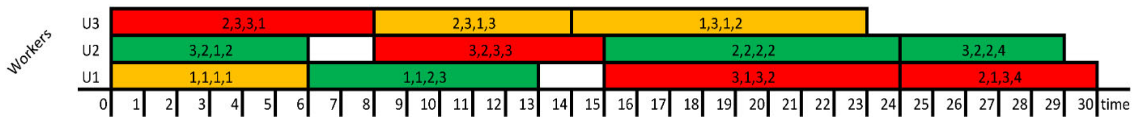



*Figure 4.4 Worker Gantt chart for illustrative example. Labels per operation (i,u,j,k).*

## 4.4 Proposed algorithm

Due to the intractability of the problem and the need for finding multiple Pareto-optimal solutions, an elitist non-dominated sorting GA (NSGA-II) is proposed. The NSGA-II is adapted to handle the requirements of DRFJSPS in which an efficient chromosome representation is presented. Three crossover operators are designed. Also, two mutation methods are developed. The following sections fully explain the elements used for the devised algorithm.

## 4.5 The elitist non-dominated sorting GA (NSGA-II)

The NSGA-II was proposed by Deb *et al.* 2002. The NSGA-II procedure, as shown in Figure 4.5, starts combining a parent population ($P_t$) with an offspring population ($Q_t$) to form the entire population ($R_t$) of size 2N. Then the population ($R_t$) is sorted according to non domination. In this case, fast-non-dominated sort (Deb *et al.* 2002) is used where ranks and fronts are identified. Solutions with rank one are the ones that belongs to the non-dominated set. Once the non-dominated sorting is done, the new population ($P_{t+1}$) of size N is formed with solutions of different non-dominated fronts (e.g. F1, F2, F3, etc.). The filling of the new population begins with the best non-dominated front (F1) and go on with the solutions of the second front (F2), and so on until there is no space in the new population. The fronts that did not fit in the new population are deleted (e.g. F4, F5 and F6). The last allowed front may not fit in the remaining slot in the new population. In this case, instead of discarding the last members of the front, the solutions are sorted according to the crowding distance operator to make the diversity.  The NSGA-II procedure is outlined in Figure 4.6.

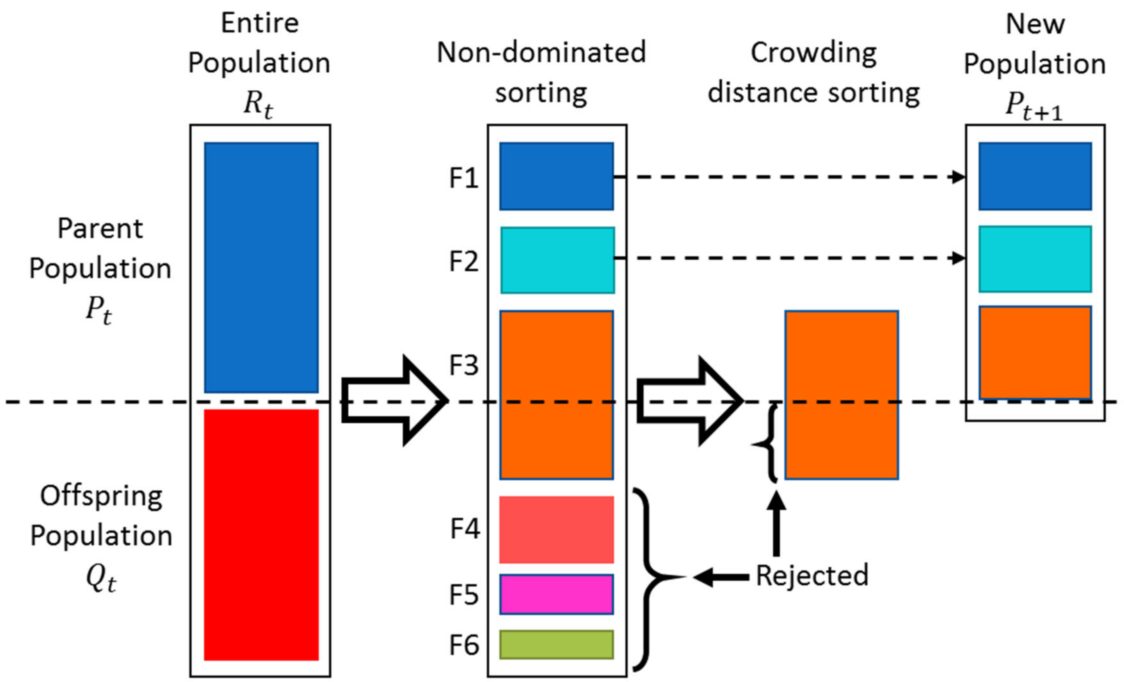



*Figure 4.5 NSGA-II schematic procedure. Adapted from(Deb et al. 2002)*

**Step 1** Create $R_t$=$P_t \cup P_t$
Perform a non-dominated sort to $R_t$
Identify fronts $F_i$ $i = 1, 2, \ldots, etc.$
**Step 2** Set $P_{t+1} = \emptyset$ and $i = 1$
Until $|P_{t+1}|+F_i < N,$ set $P_{t+1} = P_{t+1} \cup F_i$ and $i = i + 1$
**Step 3** Perform crowding-sort ($F_i$ <c) to include the most widely spread ($N - P_{t+1}|$) solutions to $P_{t+1}$
**Step 4** Create $Q_{t+1}$ from $P_{t+1}$ by using the crowded tournament selection, crossover and mutation operators.

*Figure 4.6 NSGA-II procedure.*

For the crowding sorting of the solutions of front $i$ (step 3), the crowding distance metric is calculated. Then, the population is arranged in a descending order of magnitude of the crowding distances values. In step 4, the crowded comparison operator <c compares two solutions and return the winner of the tournament. Each solution $i$, is asummed to have a non-donomination rank ($r_i$) and a local crowding distance ($d_i$). The space around $i$ that is not occupied by other solution is called $d_i$ .

To calculate $d_i$ of solution $i$, the average distance of two solutions on either side of solution $i$ along of the objectives is considered. Closest neighbours are used to for a cuboid as show in Figure 4.7. The crowding distance assignment procedure is outlined in Figure 4.8.

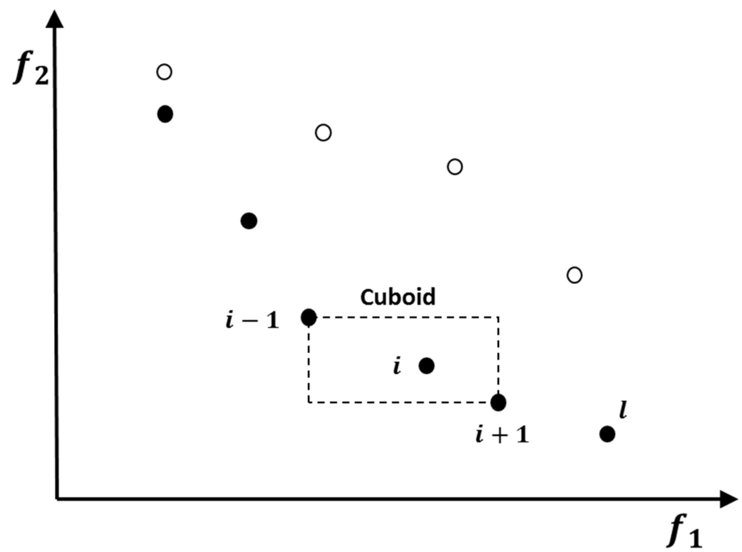



*Figure 4.7 Crowding distance calculation. Adapted from Burke and Kendall (2005).*

| |
|---|
| **Step 1** Set $l = \|F\|$ for each solution $i$<br>Set $d_i = 0$<br>**Step 2** For each objective function $m = 1, 2, \dots, M$, sort the set in ascending order of $f_m$ or find the sorted indices vector $I^m = sort\ (f_m, >)$<br>**Step 3** For $m = 1, 2, \dots, M$, set $d_{I_1^m} = d_{I_l^m} = \infty$ (boundary solution), and for the remaining solutions $j = 2, \dots, (l-1)$ assign $d_{I_j^m}$ |

*Figure 4.8 Crowding distance assignment procedure.*

In step 3 of the crowding distance assignment procedure, Equation (4.32) is used to assign the crowding distance to the solutions $j = 2$ to $(l-1)$. The index $I_j$ represents the solution of the *j*th element in the sorted list, $I_1$ symbolizes the lowest objective function and $I_l$ the highest objective function. The second part of Equation (4.32) is the difference in objective function values between two neighbouring solutions placed on the vertices of the cuboid, as per Figure 4.7. the parameters $f_m^{max}$ and $f_m^{min}$ denote the maximum and minimum values of their respective objective functions. It could be defined as the maximum and minimum of the population.

$$d_{I_j^m} = d_{I_j^m} + \frac{f_m^{(I_{j+1}^m)} - f_m^{(I_{j-1}^m)}}{f_m^{max} - f_m^{min}} \tag{4.32}$$

In the crowded tournament selection, two solutions compete (e.g. solution $i$ and solution $j$). The solution that has a better rank wins. If there is a tie between solution ranks, the solution that has a better crowding distance wins.

The flowchart of the NSGA-II for the DRFJSPS is presented in Figure 4.9

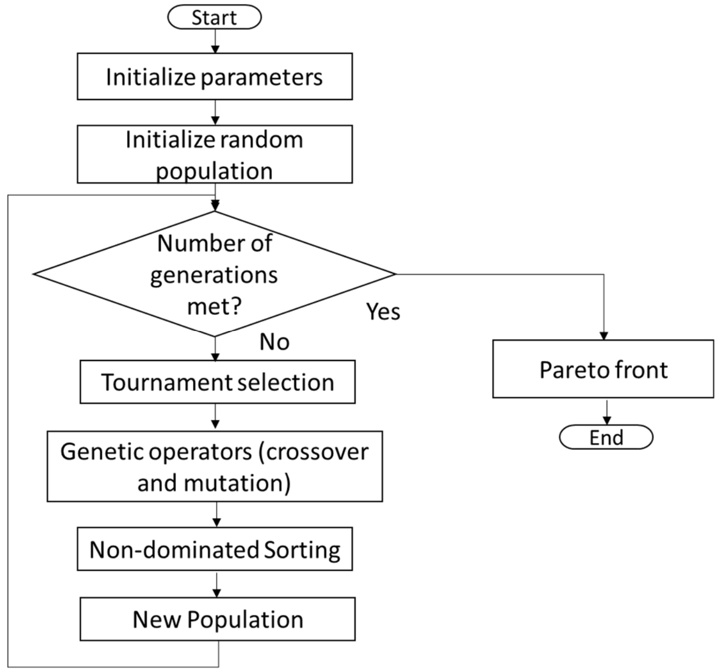



*Figure 4.9 Flowchart of NSGA-II.*

### 4.5.1 Encoding and decoding of the chromosome

The following encoding scheme is and a modification of the proposed encoding method by Zheng and Wang (2016). A feasible encoded solution for the data in Table 4.1 is presented in Figure 4.10. The solution is comprised of three vectors: Operation sequence vector (OSV), machine assignment vector (MAV) and worker assignment (WAV). The length of OSV and MAV is $\sum_j^J u_j$. For instance, the first element of OSV ($o_{2,1}$) is assigned to machine 1 ($i = 1$) and worker 2 ($u = 2$).

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{2,1}$ | $o_{3,1}$ | $o_{1,1}$ | $o_{3,2}$ | $o_{2,2}$ | $o_{2,3}$ | $o_{1,3}$ | $o_{3,3}$ | $o_{2,4}$ | $o_{3,4}$ | $o_{1,2}$ |
| MAV | 1 | 1 | 2 | 3 | 2 | 2 | 2 | 1 | 3 | 1 | 2 |
| WAV | 2 | 2 | 3 | 3 | 3 | 3 | 1 | 2 | 1 | 2 | 3 |

*Figure 4.10 A feasible solution for DRFSPS.*

The encoded solution in Figure 4.10 can be decoded with the following procedure: select an operation from OSV one at a time from left to right, then schedule the operation to the assigned machine on MAV and WAV at earliest as possible. This procedure is repeated until all the operations on OSV have been scheduled. The Gantt chart of the decoded solution is illustrated in Figure 4.11.

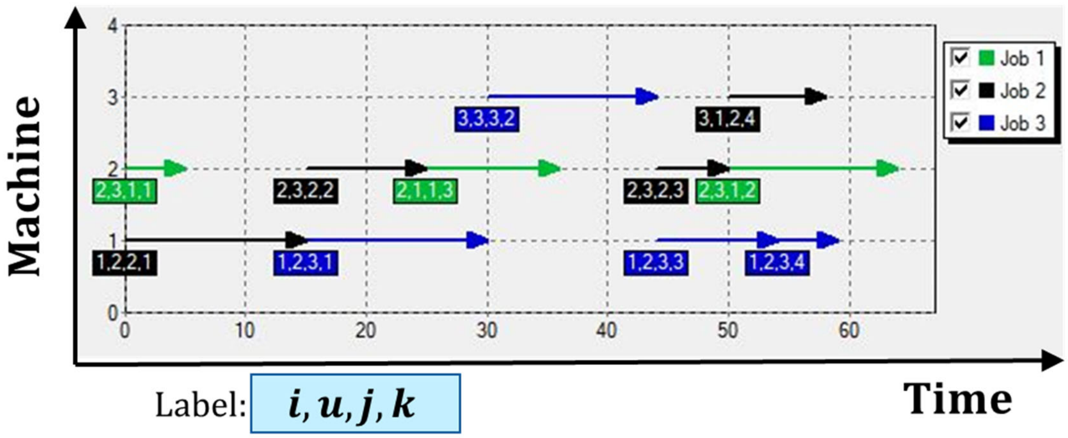



*Figure 4.11 Gantt chart of the decoded feasible solution.*

### 4.5.2 Initialization of population

NSGA-II requires the initialization of the parent population ($P_t$) of size N. The parents are created by randomly select a type of assignment (*AssigmentRule4 or AssigmentRule5*) that are described in section 4.5.2.1. Later, a random sequencing is executed that is explained in section 4.5.2.2.

#### 4.5.2.1 Initial parent population assignment

The initial machine and worker assignment are generated by two methods. The first method is called *AssigmentRule4.* It is a modification of the procedure designed by Pezzella *et al.* (2008) that finds the minimum processing time in the processing table and assigns the operation to that machine. It uses as well the approach by localization (Kacem *et al.* 2002a) where the allocated minimum time is added to the entire machine column, updating the workload.

Due to this problem is comprised of the machine and worker assignment, a modification of the previously described method is performed. In the first place, the minimum processing time is found. Then, the assignment is performed. After that, the workload update is done in all the processing times of the machine assigned. In the same way, the workload needs to be done on the columns of the worker assigned.

For instance, *AssigmentRule4* is illustrated in Table 4.3. Operation $o_{1,1}$ has the minimum processing time in the whole table at machine 2 and worker 3 ($P_{23jk}$). Operation $o_{1,1}$ is then assigned and its processing time (number underlined) is added to the entire processing times of machine 2. The same process is executed to the columns regarding worker 3. The updated processing times are shown in bold.

*Table 4.3 Initial machine assignment by AssigmentRule4 (1) (machine workload updates in bold).*

| Ope. | Machine $i = 1$ | | | Machine $i = 2$ | | | Machine $i = 3$ | | |
|---|---|---|---|---|---|---|---|---|---|
| $o_{jk}$ | $P_{11jk}$ | $P_{12jk}$ | $\boldsymbol{P_{13jk}}$ | $P_{21jk}$ | $P_{22jk}$ | $P_{23jk}$ | $P_{31jk}$ | $P_{32jk}$ | $\boldsymbol{P_{33jk}}$ |
| $o_{1,1}$ | - | 11 | - | 10 | - | **<u>5</u>** | 14 | - | **5** |
| $o_{1,2}$ | 8 | - | **11** | **14** | - | **19** | - | 8 | **20** |
| $o_{1,3}$ | 10 | - | - | **16** | **16** | **-** | - | - | - |
| $o_{2,1}$ | - | 15 | - | **20** | **12** | **-** | - | - | **10** |
| $o_{2,2}$ | 11 | - | - | **12** | - | **15** | - | - | **19** |
| $o_{2,3}$ | - | 14 | - | **13** | **11** | **11** | - | - | **13** |
| $o_{2,4}$ | 9 | 13 | **19** | - | - | **17** | 8 | - | **18** |
| $o_{3,1}$ | - | 15 | - | **15** | - | **-** | - | - | - |
| $o_{3,2}$ | - | - | **13** | - | **14** | **-** | - | 8 | **19** |
| $o_{3,3}$ | 8 | 10 | - | **18** | - | **-** | - | - | **19** |
| $o_{3,4}$ | - | 5 | **11** | - | **15** | **-** | - | 11 | - |

Once that the first assignment is done, the procedure continues looking for the second minimum processing time in the whole table. For this example, operation $o_{3,4}$ with a processing time of 5 is assigned (underlined). Then, processing times that belongs to machine 1 are updated by adding the processing time of $P_{1234}$. The same workload procedure is performed for all the columns of worker 2. The updated processing times are shown in Table 4.4. The final assignment table is presented in Table 4.5. The assignments are displayed in bold and underlined.

*Table 4.4 Initial machine assignment by AssigmentRule4 (2) (machine workload updates in bold).*

| Ope. | Machine $i = 1$ | | | Machine $i = 2$ | | | Machine $i = 3$ | | |
|---|---|---|---|---|---|---|---|---|---|
| $o_{jk}$ | $P_{11jk}$ | $\boldsymbol{P_{12jk}}$ | $P_{13jk}$ | $P_{21jk}$ | $\boldsymbol{P_{22jk}}$ | $P_{23jk}$ | $P_{31jk}$ | $\boldsymbol{P_{32jk}}$ | $P_{33jk}$ |
| $o_{1,1}$ | - | **11** | - | 10 | - | **<u>5</u>** | 14 | - | 5 |
| $o_{1,2}$ | **13** | **-** | **16** | 14 | - | 19 | - | **13** | 20 |
| $o_{1,3}$ | **15** | **-** | - | 16 | **21** | - | - | - | - |
| $o_{2,1}$ | - | **20** | - | 20 | **17** | - | - | - | 10 |
| $o_{2,2}$ | **16** | **-** | - | 12 | - | 15 | - | - | 19 |
| $o_{2,3}$ | - | **19** | - | 13 | **16** | 11 | - | - | 13 |
| $o_{2,4}$ | **14** | **18** | **24** | - | - | 17 | 8 | - | 18 |
| $o_{3,1}$ | - | **20** | - | 15 | - | - | - | - | - |
| $o_{3,2}$ | - | **-** | **18** | - | **19** | - | - | **13** | 19 |
| $o_{3,3}$ | **13** | **15** | - | 18 | - | - | - | - | 19 |
| $o_{3,4}$ | - | **<u>5</u>** | **11** | - | **15** | - | - | 11 | - |

*Table 4.5 Initial machine assignment by AssigmentRule4 (3) (machine workload updates in bold).*

| Ope. | Machine $i = 1$ | | | Machine $i = 2$ | | | Machine $i = 3$ | | |
|---|---|---|---|---|---|---|---|---|---|
| $o_{jk}$ | $P_{11jk}$ | $P_{12jk}$ | $P_{13jk}$ | $P_{21jk}$ | $P_{22jk}$ | $P_{23jk}$ | $P_{31jk}$ | $P_{32jk}$ | $P_{33jk}$ |
| $o_{1,1}$ | - | 11 | - | 10 | - | **5** | 14 | - | 5 |
| $o_{1,2}$ | 8 | - | 6 | 9 | - | **14** | - | 8 | 15 |
| $o_{1,3}$ | 10 | - | - | **11** | 11 | - | - | - | - |
| $o_{2,1}$ | - | **15** | - | 15 | 7 | - | - | - | 5 |
| $o_{2,2}$ | 11 | - | - | 7 | - | **10** | - | - | 14 |
| $o_{2,3}$ | - | 14 | - | 8 | 6 | **6** | - | - | 8 |
| $o_{2,4}$ | 9 | 13 | 14 | - | - | 12 | **8** | - | 13 |
| $o_{3,1}$ | - | **15** | - | 10 | - | - | - | - | - |
| $o_{3,2}$ | - | - | 8 | - | 9 | - | - | 8 | **14** |
| $o_{3,3}$ | 8 | **10** | - | 13 | - | - | - | - | 14 |
| $o_{3,4}$ | - | **5** | 6 | - | 10 | - | - | 11 | - |

*AssigmentRule5* uses the localization approach. It randomly permutes jobs, operations, machines and workers on the processing times table. Following this sort, the first row ($o_{jk}$) is evaluated to find the minimum processing time. The assignment is fixed and the assigned processing time is added to the entire machine and the columns with worker selected. This process is continued until all the operations have been assigned.

In Table 4.6 an example is provided. The table presents a random permutation of the jobs, operations, machines and workers. The first operation is scanned to look for the minimum $P_{iujk}$. In this example, the first row corresponds to $o_{2,3}$. The minimum processing time can be found at machine 2 and worker 2 ($P_{22jk}$). Subsequently, the assignment of $o_{2,3}$ is fixed and its processing time is added to all columns of machine 2 and worker 2.

*Table 4.6 Initial machine assignment by AssigmentRule5 (1) (machine workload updates in bold).*

| Ope. | Machine $i = 2$ | | | Machine $i = 3$ | | | Machine $i = 1$ | | |
|---|---|---|---|---|---|---|---|---|---|
| $o_{jk}$ | $P_{21jk}$ | $\boldsymbol{P_{22jk}}$ | $P_{23jk}$ | $\boldsymbol{P_{32jk}}$ | $P_{33jk}$ | $P_{31jk}$ | $P_{13jk}$ | $P_{11jk}$ | $\boldsymbol{P_{12jk}}$ |
| $o_{2,3}$ | 8 | **6** | 6 | - | 8 | - | - | - | 14 |
| $o_{2,4}$ | - | - | **18** | - | 13 | 8 | 14 | 9 | **19** |
| $o_{2,2}$ | **13** | - | **16** | - | 14 | - | - | 11 | - |
| $o_{2,1}$ | **21** | **13** | - | - | 5 | - | - | - | **21** |
| $o_{1,2}$ | **15** | - | **20** | **14** | 15 | - | 6 | 8 | - |
| $o_{1,1}$ | **16** | - | **11** | - | 5 | 14 | - | - | **17** |
| $o_{1,3}$ | **17** | **17** | - | - | - | - | - | 10 | - |
| $o_{3,1}$ | **16** | - | - | - | - | - | - | - | **21** |
| $o_{3,4}$ | - | **16** | - | **17** | - | - | 6 | - | **11** |
| $o_{3,3}$ | **19** | - | - | - | 14 | - | - | 8 | **16** |
| $o_{3,2}$ | - | **15** | - | **14** | 14 | - | 8 | - | - |

Table 4.7 shows the next step of this procedure. The row of operation $o_{2,4}$ is scanned and the minimum $P_{iujk}$ is found at machine 3 and worker 1. Its processing time is added to all columns of the machine and worker. The procedure continues until all the rows have been searched. The final assignment is shown in Table 4.8. The assignments are presented in bold and underlined.

*Table 4.7 Initial machine assignment by AssigmentRule5 (2) (machine workload updates in bold).*

| Ope. | Machine $i = 2$ | | | Machine $i = 3$ | | | Machine $i = 1$ | | |
|---|---|---|---|---|---|---|---|---|---|
| $o_{jk}$ | $\boldsymbol{P_{21jk}}$ | $P_{22jk}$ | $P_{23jk}$ | $P_{32jk}$ | $P_{33jk}$ | $\boldsymbol{P_{31jk}}$ | $P_{13jk}$ | $\boldsymbol{P_{11jk}}$ | $P_{12jk}$ |
| $o_{2,3}$ | 8 | **6** | 6 | - | 8 | - | - | - | 14 |
| $o_{2,4}$ | - | - | 18 | - | 13 | **8** | 14 | 9 | 19 |
| $o_{2,2}$ | **21** | - | 16 | - | **22** | - | - | **19** | - |
| $o_{2,1}$ | **29** | 13 | - | - | **13** | - | - | - | 21 |
| $o_{1,2}$ | **23** | - | 20 | **22** | **23** | - | 6 | **16** | - |
| $o_{1,1}$ | **24** | - | 11 | **-** | **13** | **22** | - | - | 17 |
| $o_{1,3}$ | **25** | 17 | - | **-** | - | - | - | **18** | - |
| $o_{3,1}$ | **24** | - | - | **-** | - | - | - | - | 21 |
| $o_{3,4}$ | - | 16 | - | **25** | - | - | 6 | - | 9 |
| $o_{3,3}$ | **27** | - | - | - | **22** | - | - | **16** | 16 |
| $o_{3,2}$ | - | 15 | - | **22** | **22** | - | 8 | - | - |

*Table 4.8 Initial machine assignment by AssigmentRule5 (3) (machine workload updates in bold).*

| Ope. | Machine $i = 2$ | | | Machine $i = 3$ | | | Machine $i = 1$ | | |
|---|---|---|---|---|---|---|---|---|---|
| $o_{jk}$ | $P_{21jk}$ | $P_{22jk}$ | $P_{23jk}$ | $P_{32jk}$ | $P_{33jk}$ | $P_{31jk}$ | $P_{13jk}$ | $P_{11jk}$ | $P_{12jk}$ |
| $o_{2,3}$ | 8 | **6** | 6 | - | 8 | - | - | - | 14 |
| $o_{2,4}$ | - | - | 12 | - | 13 | **8** | 14 | 9 | 13 |
| $o_{2,2}$ | 7 | - | **10** | - | 14 | - | - | 11 | - |
| $o_{2,1}$ | 15 | 7 | - | - | 5 | - | - | - | **15** |
| $o_{1,2}$ | 9 | - | **14** | 8 | 15 | - | 6 | 8 | - |
| $o_{1,1}$ | 10 | - | 5 | - | 5 | **14** | - | - | 11 |
| $o_{1,3}$ | 11 | 11 | - | - | - | - | - | **10** | - |
| $o_{3,1}$ | 10 | - | - | - | - | - | - | - | **15** |
| $o_{3,4}$ | - | 10 | - | **11** | - | - | 6 | - | 5 |
| $o_{3,3}$ | **13** | - | - | - | 14 | - | - | 8 | 10 |
| $o_{3,2}$ | - | 9 | - | 8 | **14** | - | 8 | - | - |

#### 4.5.2.2 Initial parent population sequencing

The sequencing process is executed by generating a random sequence while satisfying the precedence constraints.

### 4.5.3 Crossover operator

The crossover operator is executed to create a new offspring taking into account the parents heritage. The crossover operator is comprised of three crossover types: full-crossover, resources-crossover and sequence-crossover. The NSGA-II randomly selects one of the three crossover methods. The crossovers are defined in the following sections.

#### 4.5.3.1 Full-crossover

In this operator, two parents are crossed to generate two new solutions. In this case, a set of jobs is selected randomly. The positions of those jobs on OSV, MAV and WAV are copied from parent to offspring. In this case, the positions of the selected jobs of parent 1 are copied to offspring 1. The same procedure will be executed from parent 2 to offspring 2. Then the remaining elements in the offspring 1 are filled with the operations that do not belong to the set of jobs selected at the begging of parent 2. Those operations are copied one by one from left to right. In the same manner, offspring 2 is filled with the operations left in parent 1. This procedure is shown by an example provided in Figure 4.12. In the example, only job 3 is selected as the job to be crossed.

#### 4.5.3.2 Resources-crossover

This operator will crossover only the resources assigned in the parents. Only MAV and WAV are changed. The sequence will remain as the parents. Firstly, a random set of jobs is selected. Then, the sequence of parent 1 and 2 is copied to offspring 1 and offspring 2, respectively. After that, the resources (MAV and WAV) of the selected set of jobs are copied from parent 1 to offspring 2. The remaining elements in MAV and WAV of the offspring 1 are filled with the corresponding MAV and WAV of the OSV of parent 2. Finally, the same procedure is performed to copy the WAV and MAV elements from parent 1 to offspring 2. An example of this operator is displayed in Figure 4.13 where the selected job is job 3.

#### 4.5.3.3 Sequence-crossover

For this crossover, a set of jobs is selected randomly. The elements of the selected jobs are copied from parent 1 to offspring 1. Then, the remaining elements on OSV of offspring 1 are filled with the sequence of OSV in the parent 2 that do not belong to the jobs selected in parent 1, one by one, from left to right. Finally, the remaining elements on OSV of offspring 2 are filled with the sequence of OSV in the parent 1 that do not belong to the jobs selected in parent 2. The MAV and

RAV are fully heritage from parent to offspring.  An example is provided in Figure 4.14 where job 3 is the job selected to cross.

## Full-crossover example

Select the jobs to be crossed: Job 3

**Parent 1**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{1,1}$ | $o_{1,2}$ | $o_{1,3}$ | $\boldsymbol{o_{3,1}}$ | $\boldsymbol{o_{3,2}}$ | $\boldsymbol{o_{3,3}}$ | $\boldsymbol{o_{3,4}}$ | $o_{2,1}$ | $o_{2,2}$ | $o_{2,3}$ | $o_{2,4}$ |
| MAV | 2 | 3 | 2 | **1** | **3** | **1** | **1** | 2 | 2 | 1 | 3 |
| WAV | 3 | 2 | 1 | **2** | **2** | **1** | **2** | 2 | 1 | 2 | 1 |

**Offspring 1**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{2,1}$ | $o_{1,1}$ | $o_{1,2}$ | $\boldsymbol{o_{3,1}}$ | $\boldsymbol{o_{3,2}}$ | $\boldsymbol{o_{3,3}}$ | $\boldsymbol{o_{3,4}}$ | $o_{2,2}$ | $o_{1,3}$ | $o_{2,3}$ | $o_{2,4}$ |
| MAV | 2 | 1 | 3 | **1** | **3** | **1** | **1** | 2 | 2 | 1 | 3 |
| WAV | 2 | 2 | 3 | **2** | **2** | **1** | **2** | 1 | 1 | 2 | 1 |

**Parent 2**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{2,1}$ | $\boldsymbol{o_{3,1}}$ | $o_{1,1}$ | $o_{1,2}$ | $o_{2,2}$ | $\boldsymbol{o_{3,2}}$ | $o_{1,3}$ | $\boldsymbol{o_{3,3}}$ | $o_{2,3}$ | $\boldsymbol{o_{3,4}}$ | $o_{2,4}$ |
| MAV | 2 | **2** | 1 | 3 | 2 | **1** | 2 | **3** | 1 | **2** | 3 |
| WAV | 2 | **1** | 2 | 3 | 1 | **3** | 1 | **3** | 2 | **2** | 1 |

**Offspring 2**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{1,1}$ | $\boldsymbol{o_{3,1}}$ | $o_{1,2}$ | $o_{1,3}$ | $o_{2,1}$ | $\boldsymbol{o_{3,2}}$ | $o_{2,2}$ | $\boldsymbol{o_{3,3}}$ | $o_{2,3}$ | $\boldsymbol{o_{3,4}}$ | $o_{2,4}$ |
| MAV | 2 | **2** | 3 | 2 | 2 | **1** | 2 | **3** | 1 | **2** | 3 |
| WAV | 3 | **1** | 2 | 1 | 2 | **3** | 1 | **3** | 2 | **2** | 1 |

**Parent 1**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{1,1}$ | $o_{1,2}$ | $o_{1,3}$ | $o_{3,1}$ | $o_{3,2}$ | $o_{3,3}$ | $o_{3,4}$ | $o_{2,1}$ | $o_{2,2}$ | $o_{2,3}$ | $o_{2,4}$ |
| MAV | 2 | 3 | 2 | 1 | 3 | 1 | 1 | 2 | 2 | 1 | 3 |
| WAV | 3 | 2 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 2 | 1 |

*Figure 4.12 Full-Crossover example.*

## Resources-crossover example

Select the jobs to be crossed: Job 3

Parent 1

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{1,1}$ | $o_{1,2}$ | $o_{1,3}$ | $\boldsymbol{o_{3,1}}$ | $\boldsymbol{o_{3,2}}$ | $\boldsymbol{o_{3,3}}$ | $\boldsymbol{o_{3,4}}$ | $o_{2,1}$ | $o_{2,2}$ | $o_{2,3}$ | $o_{2,4}$ |
| MAV | 2 | 3 | 2 | **1** | **3** | **1** | **1** | 2 | 2 | 1 | 3 |
| WAV | 3 | 2 | 1 | **2** | **2** | **1** | **2** | 2 | 1 | 2 | 1 |

Offspring 1

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{1,1}$ | $o_{1,2}$ | $o_{1,3}$ | $\boldsymbol{o_{3,1}}$ | $\boldsymbol{o_{3,2}}$ | $\boldsymbol{o_{3,3}}$ | $\boldsymbol{o_{3,4}}$ | $o_{2,1}$ | $o_{2,2}$ | $o_{2,3}$ | $o_{2,4}$ |
| MAV | 1 | 3 | 2 | **1** | **3** | **1** | **1** | 2 | 2 | 1 | 3 |
| WAV | 2 | 3 | 1 | **2** | **2** | **1** | **2** | 2 | 1 | 2 | 1 |

Parent 2

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{2,1}$ | $\boldsymbol{o_{3,1}}$ | $o_{1,1}$ | $o_{1,2}$ | $o_{2,2}$ | $\boldsymbol{o_{3,2}}$ | $o_{1,3}$ | $\boldsymbol{o_{3,3}}$ | $o_{2,3}$ | $\boldsymbol{o_{3,4}}$ | $o_{2,4}$ |
| MAV | 2 | **2** | 1 | 3 | 2 | **1** | 2 | **3** | 1 | **2** | 3 |
| WAV | 2 | **1** | 2 | 3 | 1 | **3** | 1 | **3** | 2 | **2** | 1 |

Offspring 2

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{2,1}$ | $\boldsymbol{o_{3,1}}$ | $o_{1,1}$ | $o_{1,2}$ | $o_{2,2}$ | $\boldsymbol{o_{3,2}}$ | $o_{1,3}$ | $\boldsymbol{o_{3,3}}$ | $o_{2,3}$ | $\boldsymbol{o_{3,4}}$ | $o_{2,4}$ |
| MAV | 2 | **2** | 2 | 3 | 2 | **1** | 2 | **3** | 1 | **2** | 3 |
| WAV | 2 | **1** | 3 | 2 | 1 | **3** | 1 | **3** | 2 | **2** | 1 |

Parent 1

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{1,1}$ | $o_{1,2}$ | $o_{1,3}$ | $o_{3,1}$ | $o_{3,2}$ | $o_{3,3}$ | $o_{3,4}$ | $o_{2,1}$ | $o_{2,2}$ | $o_{2,3}$ | $o_{2,4}$ |
| MAV | 2 | 3 | 2 | 1 | 3 | 1 | 1 | 2 | 2 | 1 | 3 |
| WAV | 3 | 2 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 2 | 1 |

*Figure 4.13 Crossover example (resources).*

# Sequence-crossover example

Select the jobs to be crossed: Job 3

**Parent 1**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{1,1}$ | $o_{1,2}$ | $o_{1,3}$ | $\mathbf{o_{3,1}}$ | $\mathbf{o_{3,2}}$ | $\mathbf{o_{3,3}}$ | $\mathbf{o_{3,4}}$ | $o_{2,1}$ | $o_{2,2}$ | $o_{2,3}$ | $o_{2,4}$ |
| MAV | 2 | 3 | 2 | 1 | 3 | 1 | 1 | 2 | 2 | 1 | 3 |
| WAV | 3 | 2 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 2 | 1 |

**Offspring 1**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{2,1}$ | $o_{1,1}$ | $o_{1,2}$ | $\mathbf{o_{3,1}}$ | $\mathbf{o_{3,2}}$ | $\mathbf{o_{3,3}}$ | $\mathbf{o_{3,4}}$ | $o_{2,2}$ | $o_{1,3}$ | $o_{2,3}$ | $o_{2,4}$ |
| MAV | 2 | 2 | 3 | 1 | 3 | 1 | 1 | 2 | 2 | 1 | 3 |
| WAV | 2 | 3 | 2 | 2 | 2 | 1 | 2 | 1 | 1 | 2 | 1 |

**Parent 2**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{2,1}$ | $\mathbf{o_{3,1}}$ | $o_{1,1}$ | $o_{1,2}$ | $o_{2,2}$ | $\mathbf{o_{3,2}}$ | $o_{1,3}$ | $\mathbf{o_{3,3}}$ | $o_{2,3}$ | $\mathbf{o_{3,4}}$ | $o_{2,4}$ |
| MAV | 2 | 2 | 1 | 3 | 2 | 1 | 2 | 3 | 1 | 2 | 3 |
| WAV | 2 | 1 | 2 | 3 | 1 | 3 | 1 | 3 | 2 | 2 | 1 |

**Offspring 2**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{1,1}$ | $\mathbf{o_{3,1}}$ | $o_{1,2}$ | $o_{1,3}$ | $o_{2,1}$ | $\mathbf{o_{3,2}}$ | $o_{2,2}$ | $\mathbf{o_{3,3}}$ | $o_{2,3}$ | $\mathbf{o_{3,4}}$ | $o_{2,4}$ |
| MAV | 1 | 2 | 3 | 2 | 2 | 1 | 2 | 3 | 1 | 2 | 3 |
| WAV | 2 | 1 | 3 | 1 | 2 | 3 | 1 | 3 | 2 | 2 | 1 |

**Parent 1**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{1,1}$ | $o_{1,2}$ | $o_{1,3}$ | $o_{3,1}$ | $o_{3,2}$ | $o_{3,3}$ | $o_{3,4}$ | $o_{2,1}$ | $o_{2,2}$ | $o_{2,3}$ | $o_{2,4}$ |
| MAV | 2 | 3 | 2 | 1 | 3 | 1 | 1 | 2 | 2 | 1 | 3 |
| WAV | 3 | 2 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 2 | 1 |

*Figure 4.14 Crossover example (sequence).*

### 4.5.4 Mutation operator

Mutation is a procedure that is employed to generate and explore new solutions. This mechanism will try to avoid to get stuck in a local optimal. This procedure is executed based on a mutation probability $M_p$. Two mutations operator are developed. The selection of the mutation type is randomly selected. The operators are described in sections 4.5.1 and 4.5.2.

#### 4.5.4.1 Resources-mutation

The resource-mutation operator is generated by changing a certain number of elements in MAV and WAV simultaneously. First, the total number of mutations $m_t$ is randomly calculated. Due to the number of elements mutated need to be low, a maximum of $\lceil \sum_j^J u_j * 0.3 \rceil$ mutations is allowed. Secondly, a random operation is selected and the worker and machine assignment are randomly selected out of the set of capable resources. Finally , the procedure is executed $m_t$ times. An example is provided in Figure 4.15 where the selected operations is $o_{3,1}$ and its machine and worker assignment is changed.

**Sequence-mutation example**

Select the operation to be mutated: $o_{3,1}$

Solution 1

| OSV | $o_{1,1}$ | $o_{1,2}$ | $o_{1,3}$ | $\boldsymbol{o_{3,1}}$ | $o_{3,2}$ | $o_{3,3}$ | $o_{3,4}$ | $o_{2,1}$ | $o_{2,2}$ | $o_{2,3}$ | $o_{2,4}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MAV | 2 | 3 | 2 | **1** | 3 | 1 | 1 | 2 | 2 | 1 | 3 |
| WAV | 3 | 2 | 1 | **2** | 2 | 1 | 2 | 2 | 1 | 2 | 1 |

Mutation 1

| OSV | $o_{1,1}$ | $o_{1,2}$ | $o_{1,3}$ | $\boldsymbol{o_{3,1}}$ | $o_{3,2}$ | $o_{3,3}$ | $o_{3,4}$ | $o_{2,1}$ | $o_{2,2}$ | $o_{2,3}$ | $o_{2,4}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MAV | 1 | 3 | 2 | **2** | 3 | 1 | 1 | 2 | 2 | 1 | 3 |
| WAV | 2 | 3 | 1 | **1** | 2 | 1 | 2 | 2 | 1 | 2 | 1 |

*Figure 4.15 Mutation example (resources).*

#### 4.5.4.2 Sequence -mutation

This operator executes pairwise exchanges by randomly selecting two operations. The operations are swapped. This procedure is performed a total number of mutations $m_t$ which is randomly calculated. The maximum number of mutations is set to $\left\lceil \sum_j^J u_j * 0.3 \right\rceil$. The example in Figure 4.16 perform a pairwise exchange of operators $o_{3,2}$ and $o_{3,3}$. These exchanges can produce an infeasible solution a repair mechanism is presented in Figure 4.17.

## Mutation example (Sequence)

Select the operation to be mutated: $o_{3,2}$ to $o_{3,3}$

Solution 1

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{1,1}$ | $o_{1,2}$ | $o_{1,3}$ | $o_{3,1}$ | $\boldsymbol{o_{3,2}}$ | $\boldsymbol{o_{3,3}}$ | $o_{3,4}$ | $o_{2,1}$ | $o_{2,2}$ | $o_{2,3}$ | $o_{2,4}$ |
| MAV | 2 | 3 | 2 | 1 | **3** | **1** | 1 | 2 | 2 | 1 | 3 |
| WAV | 3 | 2 | 1 | 2 | **2** | **1** | 2 | 2 | 1 | 2 | 1 |

Mutation 1

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSV | $o_{1,1}$ | $o_{1,2}$ | $o_{1,3}$ | $o_{3,1}$ | $\boldsymbol{o_{3,3}}$ | $\boldsymbol{o_{3,2}}$ | $o_{3,4}$ | $o_{2,1}$ | $o_{2,2}$ | $o_{2,3}$ | $o_{2,4}$ |
| MAV | 2 | 3 | 2 | 1 | **1** | **3** | 1 | 2 | 2 | 1 | 3 |
| WAV | 3 | 2 | 1 | 2 | **1** | **2** | 2 | 2 | 1 | 2 | 1 |

*Figure 4.16 Mutation example (sequence).*

```
Begin
  repair ← true
  While repair = true Do
    For r=1 to ∑_j^J u_j − 1 Do
      For s=1 to ∑_j^J u_j Do
        If OSV(r) and OSV(s) follow precedence relationship Then
          repair ← false
        else
          repair ← true
          Swap positions rth and sth on OSV, MAV and WAV
        End If
      End Do
    End Do
  End Do
End
```

*Figure 4.17 Repair mechanism for sequence-mutation.*

## 4.6 Summary

In this chapter, the DRFJSPS has been defined. The different assumption that involves the worker integration to the FJSPS has been underlined. The multi-objective approach that deals with the minimization of makespan, maximal worker workload and weighted tardiness are presented. Due to the intractability of the problem, a multi-objective genetic algorithm is devised. An encoding scheme is generated to fit the chromosome of the algorithm. In the same manner, two new assignments procedures to create the initial population are explained. Similarly, the algorithm incorporates new genetic operators. Three new crossovers and two mutation method are introduced. The experimentation portion of the proposed approach is discussed in Numerical experiments CHAPTER 7.

# CHAPTER 5. FLEXIBLE JOB-SHOP SCHEDULING PROBLEM WITH SEQUENCING AND PROCESS PLAN FLEXIBILITY (FJSP-2F)

The increased product variety and increased competition is a characteristic of Today's manufacturing environment. Manufactures has to adapt to the unpredictable demand quickly. The possibility of having alternative process plans can give a competitive advantage that can lead to a thriving industry. This chapter presents the development of a mathematical formulation for the FJSPS with the integration of process plan flexibility and total processing cost.

## 5.1 Introduction

Process planning and scheduling are two crucial functions in the manufacturing environment. The former determines the detailed manufacturing process and the required technological resources to achieve a product. The latter defines the sequence of operations, in the particular case of the FJSP, it assigns the operations to the machines. Moreover, these functions are implemented in a sequential manner; firstly, process planning is executed, and finally, scheduling is performed.

Usually, it is assumed that only process plan is available for a job. This assumption is often not real (Kim and Egbelu 1999). In order to take advantage of the different flexibilities in today's modern manufacturing environments, alternative process plan can be generated. A process plan is generated without information of the current manufacturing resources, and it lacks consideration of scheduling objectives. The process plan generated could also not be feasible due to the present manufacturing scenario. Maintaining multiple process plans in these manufacturing systems presents a solution to this problem. In this way, the shop can be adapted to react to the current product mix and machine failures.

Another important element when selecting an alternative process plan is that integration of a processing cost. Better schedules can be obtained by increasing costs (Brandimarte and Calderini 1995). This is a fundamental element that needs to be considered when selecting a process plan.

As mentioned in CHAPTER 3 and CHAPTER 4, manufacturers have commitments to accomplish like due dates. In case of failing a due date, a penalty is applied by the customer. The penalty could impact generously on the production profits. Weighted tardiness is an objective that deals with this typical scenario.

In conclusion, the new manufacturing environments enable the capacity to provide with different flexibilities that allow to efficiently to react to changes in the demand and the current shop status. Now, the managers need to deal with the selection of a process plan that captures the current manufacturing scenario, decide resource allocation and deliver the jobs on time. This situation creates a very complex problem that needs to be studied. Hence, a multi-objective mathematical formulation for the FJSP with process plan and sequencing flexibility is developed.

## 5.2 Problem description

The flexible job-shop scheduling problem with sequencing and process plans flexibility (FJSP-2F) considers $n$ jobs that have to be processed in $m$ machines. Each job $j$ has a $b_j$ total number of alternative process plans. Each process plan $p$ contains a total of $u_{jp}$ operations to process job $j$. The total number of $u_{jp}$ operations is not necessary the same in each $p$ plan. Only one process plan $p$ needs to be selected to process job $j$. Precedence between the operations are given by an arbitrary directed acyclic graph. There is a cost $g_{ijkp}$ to process an operation $O_{jkp}$ on a certain machine $i$. The objectives in the multi-objective FJSP-2F with sequencing flexibility are the minimization of the following criteria:

- Makespan
- Total processing cost
- Weighted tardiness

The following assumptions are proposed for the FJSP-2F:

- All the machines are available at time zero.
- Each machine can process only one operation at a time on any job.
- Each operation can be performed only once on one machine, and its sequence is respected for every job.
- Transportation time is not considered.
- Setup time of an operation is included in the processing time.
- The operations of different jobs do not have precedence constraints.
- Job preemption is not allowed.
- An operation of any job cannot be processed until its preceding operations are completed.
- Alternative process plans with sequencing flexibility are given for each job.
- There is a cost to process an operation of a certain job in a certain machine.
- Processing cost is given in advance.
- The processing time corresponding to the jobs, operations, machines, and process plan, is given in advance.

## 5.3 Mathematical formulation

A mixed integer linear programming model is presented to tackle the FJSP-2F. The notation required for the model is introduced below.

**Parameters and indexes**

$n$: Number of jobs
$j$: Index of jobs $(1, \dots, n)$
$m$: Number of machines
$i$: Index of machines $(1, \dots, m)$
$b_j$: Total number of alternative process plan for job $j$
$p$: Index of process plans $(1, \dots, b_j)$
$u_{jp}$: Total number of operations of process plan $p$ job $j$
$k$: Index of operations $(1, \dots, u_{jp})$
$O_{jkp}$: Operation $k$ of process plan $p$ of job $j$
$P_{ijkp}$: Processing time of operation $k$ of process plan $p$ of job $j$ on machine $i$
$e_{jkk'p}$: $\begin{cases} 1 \text{ if operation } k \text{ precedes operation } k' \text{ of process plan } p \text{ of job } j \\ 0 \text{ otherwise} \end{cases}$
$a_{ijkp}$: $\begin{cases} 1 \text{ if machine } i \text{ can process operation } k \text{ of process plan } p \text{ of job } j \\ 0 \text{ otherwise} \end{cases}$
$g_{ijkp}$: Cost of processing operation $k$ of process plan $p$ of job $j$
$d_j$: due date of job $j$
$W_j$: the weight of job $j$

**Decision variables**

$x_{ijkp}$: $\begin{cases} 1 \text{ if operation } k \text{ of process plan } p \text{ of job } j \text{ is processed on machine } i \\ 0 \text{ otherwise} \end{cases}$
$h_{jp}$: $\begin{cases} 1 \text{ if process plan } p \text{ is selected for job } j \\ 0 \text{ otherwise} \end{cases}$
$C_{ijkp}$: Completion time of operation $k$ of process plan $p$ of job $j$ on machine $i$
$T_j$: tardiness of job $j$
$c_j$: Completion time of job $j$
$G$: Total processing cost
$Y_{ijkk'p}$: $\begin{cases} 1 \text{ if the operation } k \text{ precedes operation } k' \text{ of process plan } p \text{ of job } j \text{ on machine } i \\ 0 \text{ otherwise} \end{cases}$

$Q_{ii'jkk'p}$: $\begin{cases} 1 \text{ if operation } k \text{ on machine } i \text{ precedes operation } k' \text{ in machine } i' \text{ of job } j \text{ of} \\ \quad \text{process plan } p \\ 0 \text{ otherwise} \end{cases}$

$Z_{ijj'kk'pp'}$: $\begin{cases} 1 \text{ if operation } k \text{ of process plan } p \text{ of job } j \text{ precedes operation } k' \text{ of process plan} \\ \quad p' \text{ of job } j' \text{ on machine } i \\ 0 \text{ otherwise} \end{cases}$

$$min\ f_1 = c_{max} \tag{5.1}$$

$$min\ f_2 = G \tag{5.2}$$

$$min\ f_3 = \sum_{j=1}^{n} W_j T_j \tag{5.3}$$

$$\sum_{i=1}^{m} x_{ijkp} = h_{jp} \qquad \forall\, j,k,p \tag{5.4}$$

$$\sum_{p=1}^{b_j} h_{jp} = 1 \qquad \forall\, j \tag{5.5}$$

$$C_{i'jk'p} \geq C_{ijkp} + x_{i'jk'p} \cdot P_{i'jk'p} - M\left(1 - x_{i'jk'p}\right) \forall\, i,i',j,k \neq k',p\colon e_{jkk'p} = 1, a_{ijkp} = 1, \quad a_{i'jk'p} = 1 \tag{5.6}$$

$$C_{ijk'p} \geq C_{ijkp} + P_{ijk'p} - M\left(3 - Y_{ijkk'p} - x_{ijkp} - x_{ijk'p}\right) \forall\, i,j,k \neq k',p \tag{5.7}$$

$$C_{ijkp} \geq C_{ijk'p} + P_{iujk} - M\left(2 + Y_{ijkk'p} - x_{ijkp} - x_{ijk'p}\right) \forall\, i,j,k \neq k',p \tag{5.8}$$

$$C_{i'jk'p} \geq C_{ijkp} + P_{i'jk'p} - M\left(3 - Q_{ii'jkk'p} - x_{ijkp} - x_{i'jk'p}\right) \forall\, i \neq i',j,k \neq k',p \tag{5.9}$$

$$C_{ijkp} \geq C_{i'jk'p} + P_{ijkp} - M\left(2 + Q_{ii'jkk'p} - x_{ijkp} - x_{i'jk'p}\right) \forall\ i \neq i',j,k \neq k',p \tag{5.10}$$

$$C_{ij'k'p'} \geq C_{ijkp} + P_{ij'k'p'} - M\left(3 - Z_{ijj'kk'pp'} - x_{ijkp} - x_{ij'k'p'}\right) \forall\, i,j \neq j',k,k',p,p' \tag{5.11}$$

$$C_{ijkp} \geq C_{ij'k'p} + P_{ijkp} - M\left(2 + Z_{ijj'kk'pp'} - x_{ijkp} - x_{ij'k'p'}\right) \forall\, i,j \neq j',k,k',p,p' \tag{5.12}$$

$$x_{ijkp} \leq a_{ijkp} \ \forall\, i,j,k,p \tag{5.13}$$

$$C_{ijkp} - P_{ijkp} \geq\ -M\left(1 - x_{ijkp}\right) \forall\, i,j,k,p \tag{5.14}$$

$$C_{ijkp} \leq M\left(x_{ijkp}\right) \forall\, i,j,k,p \tag{5.15}$$

$$c_{max} \geq C_{ijkp} \ \forall\, i,j,k,p \tag{5.16}$$

$$G = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k}^{u_{jp}} \sum_{p=1}^{b_j} x_{ijkp}\, g_{ijkp} \tag{5.17}$$

$$T_j \geq c_j - d_j \ \forall\, j \tag{5.18}$$

$$T_j \geq 0\ \forall \tag{5.19}$$

$$c_j \geq C_{ijkp} \ \forall\, i,u,j,k \tag{5.20}$$

$$C_{ijkp} \geq 0 \forall\, i,j,u,k \tag{5.21}$$

$$x_{ijkp}, Y_{ijkk'p}, Q_{ii'jkk'p}, Z_{ijj'kk'p} \ \in \{0,1\} \ \forall\, i,i',j,j'k,k',p \tag{5.22}$$

Eq. (5.1), Eq. (5.2) and Eq.(5.3) defines makespan, total processing cost and weighted tardiness that are the objectives to be minimized. Constraint set (5.4) ensures that the only operation is selected from a set of available machines of process plan $p$. Constraint set (5.5) assures that only one process plan $p$ is selected for job $j$. Constraint set (5.6) states the precedence among the operations of a job. The disjunctive constraint set (Eq. (5.7) and (5.8)) ensures that only one operation can be processed in a machine at a time. Constraint set (Eq.(5.9) and (5.10)) avoids the overlapping of operations in different machines for the same job and process plan at a time. Constraint set (Eq.(5.11) and Eq.(5.12)) avoids the overlapping of operations different jobs and process plans for the same machine at a time. Constraint set Eq.(5.13) specifies the feasibility of the machine assignment. Constraint set (Eq. (5.14) and Eq.(5.15)) defines the completion time when a machine tool has been assigned. Constraint set Eq.(5.16) defines the makespan. Eq. (5.17) specifies the total processing cost. Constraint set (Eq.(5.18) and Eq.(5.19)) defines the tardiness. Constraint set (Eq.(5.20), Eq.(5.21) and Eq.(5.22)) defines the types of variable.

## 5.4 Illustrative example

In Table 5.1, an illustrative instance is presented for the FJSP-2F with two jobs, three machines. Precedence relationships for this illustrative example can be found in Figure 5.1. The associated processing times and costs are provided in Table 5.1.

*Table 5.1 Processing times and cost of an illustrative instance.*

| Processing times | | | | | Processing costs | | | | |
|---|---|---|---|---|---|---|---|---|---|
| job | op j,k,p | Machine 1 | Machine 2 | Machine 3 | job | op j,k,p | Machine 1 | Machine 2 | Machine 3 |
| | 1,1,1 | 15 | - | 12 | | 1,1,1 | 3 | 0 | 7 |
| | 1,2,1 | - | 14 | - | | 1,2,1 | 0 | 1 | 0 |
| | 1,3,1 | 8 | 10 | 11 | | 1,3,1 | 7 | 8 | 3 |
| 1 | 1,1,2 | 14 | - | 9 | 1 | 1,1,2 | 5 | 0 | 1 |
| | 1,2,2 | 12 | - | - | | 1,2,2 | 6 | 0 | 0 |
| | 1,3,2 | 13 | - | 14 | | 1,3,2 | 5 | 0 | 10 |
| | 1,4,2 | 10 | 6 | - | | 1,4,2 | 2 | 10 | 0 |
| | 2,1,1 | - | 13 | - | | 2,1,1 | 0 | 3 | 0 |
| | 2,2,1 | 12 | - | - | | 2,2,1 | 8 | 0 | 0 |
| | 2,3,1 | 10 | 8 | - | | 2,3,1 | 7 | 7 | 0 |
| 2 | 2,1,2 | - | 5 | 14 | 2 | 2,1,2 | 0 | 8 | 6 |
| | 2,2,2 | 12 | 11 | 6 | | 2,2,2 | 4 | 4 | 1 |
| | 2,3,2 | 9 | - | - | | 2,3,2 | 3 | 0 | 0 |
| | 2,4,2 | - | - | 6 | | 2,4,2 | 0 | 0 | 3 |
| | 3,1,1 | - | - | 15 | | 3,1,1 | 0 | 0 | 7 |
| | 3,2,1 | 9 | - | 10 | | 3,2,1 | 3 | 0 | 9 |
| 3 | 3,3,1 | - | 12 | 7 | 3 | 3,3,1 | 0 | 3 | 5 |
| | 3,1,2 | 5 | 5 | - | | 3,1,2 | 4 | 9 | 0 |
| | 3,2,2 | 5 | 8 | 7 | | 3,2,2 | 5 | 1 | 2 |
| | 3,3,2 | 11 | - | 15 | | 3,3,2 | 8 | 0 | 5 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 3,4,2 | 14 | 9 | - | 3,4,2 | 5 | 2 | 0 |
| 3,1,3 | 7 | - | 8 | 3,1,3 | 6 | 0 | 1 |
| 3,2,3 | - | 13 | 8 | 3,2,3 | 0 | 6 | 8 |
| 3,3,3 | 13 | - | - | 3,3,3 | 4 | 0 | 0 |
| 3,4,3 | 14 | 5 | 11 | 3,4,3 | 8 | 4 | 6 |

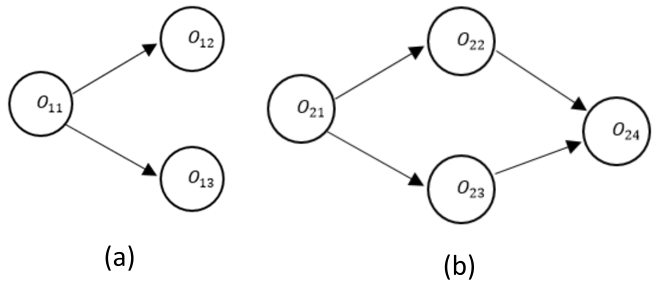



*Figure 5.1 Operation precedence.*

In order to integrate the multiple objectives, the weighted sum approach has been used, and it can be found in Eq. (4.31). The weights $w_1'$, $w_2'$ and $w_3'$ are input data provided by the user.

$$min\ f' = w_1'f_1 + w_2'f_2 + w_3'f_3 \tag{5.23}$$

Eq. (4.31) employs three weights ($w_1' = .3$, $w_2' = .3$ and $w_3' = .4$).

Xpress Optimizer 25.01.05 algebraic model language and optimizer have been used to code the mathematical model. The optimal solution has been found after 4.2 seconds. The value of different objectives is presented in Table 5.2. The Gantt chart in Figure 5.2 shows the optimal solution for this example.

*Table 5.2 The experimental results for the weighted tardiness data.*

| Objective | Value |
|---|---|
| Makespan | 36 |
| Processing cost | 45 |
| Weighted tardiness | 78 |

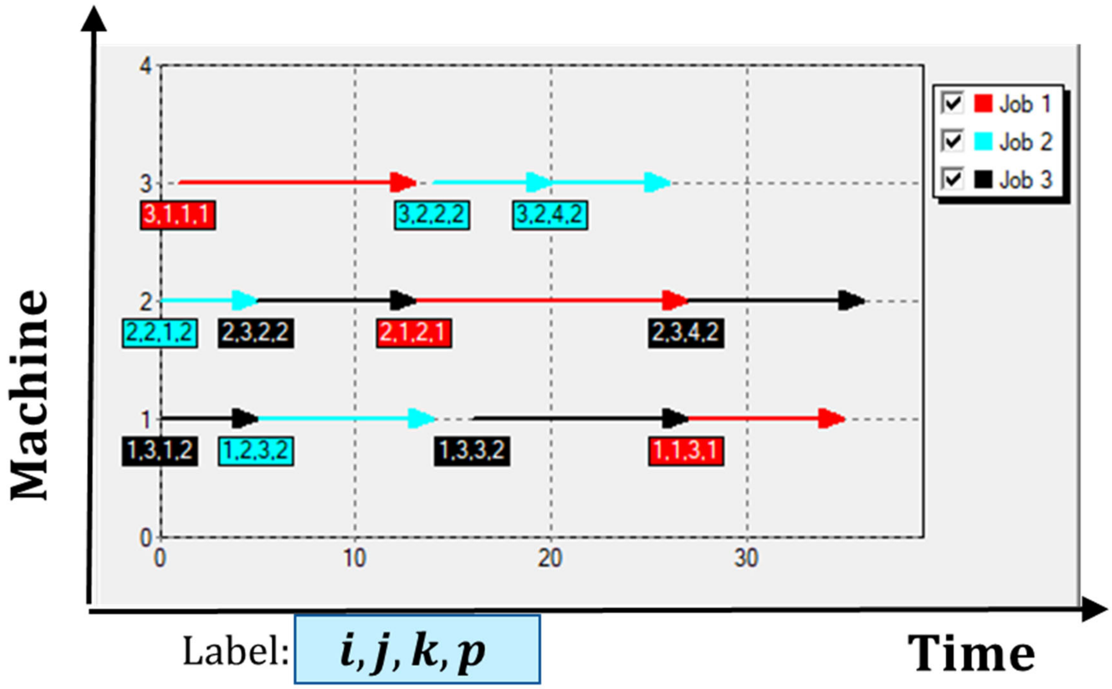



*Figure 5.2 Gantt chart of the illustrative example.*

## 5.5 Summary

In this chapter, a novel multi-objective mathematical formulation for the FJSP-2F. The integration of alternative process plans to FJSPS, gives the advantage to better to unpredictable changes on the shop floor. In addition, the model integrates processing cost, makespan and weighted tardiness as objectives to be minimized. Numerical experimentation of this problem is provided in CHAPTER 7.

# CHAPTER 6. APPLICATION IN METAL CUTTING INDUSTRY: CASE STUDY AND DECISION SUPPORT SYSTEM

This chapter shows a case study from a local carbide tool and inserts vendor and fabricator in Ontario, Canada. Currently, the vendor has increased his product mix due to the introduction of new customers. The company has been developing and growing to satisfy customer needs; one of the most significant changes in the company is the acquisition of newer and more advanced manufacturing resources. Nowadays, the company has a mix of old and new machines. The different machines have different manufacturing capacities. For instance, some new machines are faster and can handle a larger order when compared with the old ones.

Some customers require to deliver their products on a specific date. A penalty is applied if the manufacturer does not follow this policy. In the same manner, the company needs to expedite specific jobs that are rush orders from the customer. For the company, customer satisfaction is a priority, and the manager wants to employ all the available to reduce time-to-market. The massive increase of the product mix has made the scheduling process a very complex activity. This scenario is a classic example and application of the weighted tardiness minimization objective function.

The following processes are the most common manufacturing process for most of the jobs in the company:

- Honing
- Heat treatment
- Face grinding
- Periphery grinding
- Drilling
- Surface grinding
- Coating

Notably, the precedence relations between the operations of specific products are given by a directed acyclic graph. The manager wants to take this situation as an advantage when the scheduling process is performed.

Three cases have been provided. Case I is an example of 10 jobs and 15 machines with a total of 63 operations. Case II is an example of 5 jobs and 12 machines with a total of 25 operations. Case

III is an example of 8 jobs and 14 machines with a total of 44 operations. The data for the case I, II and II are presented in APPENDIX A, Table A.1, Table A.3 and Table A.4 respectively. Operations precedence graphs can be found in Figure 6.1. The graphs used for Case I, II and III are presented in Table 6.1, Table 6.2 and Table 6.3.

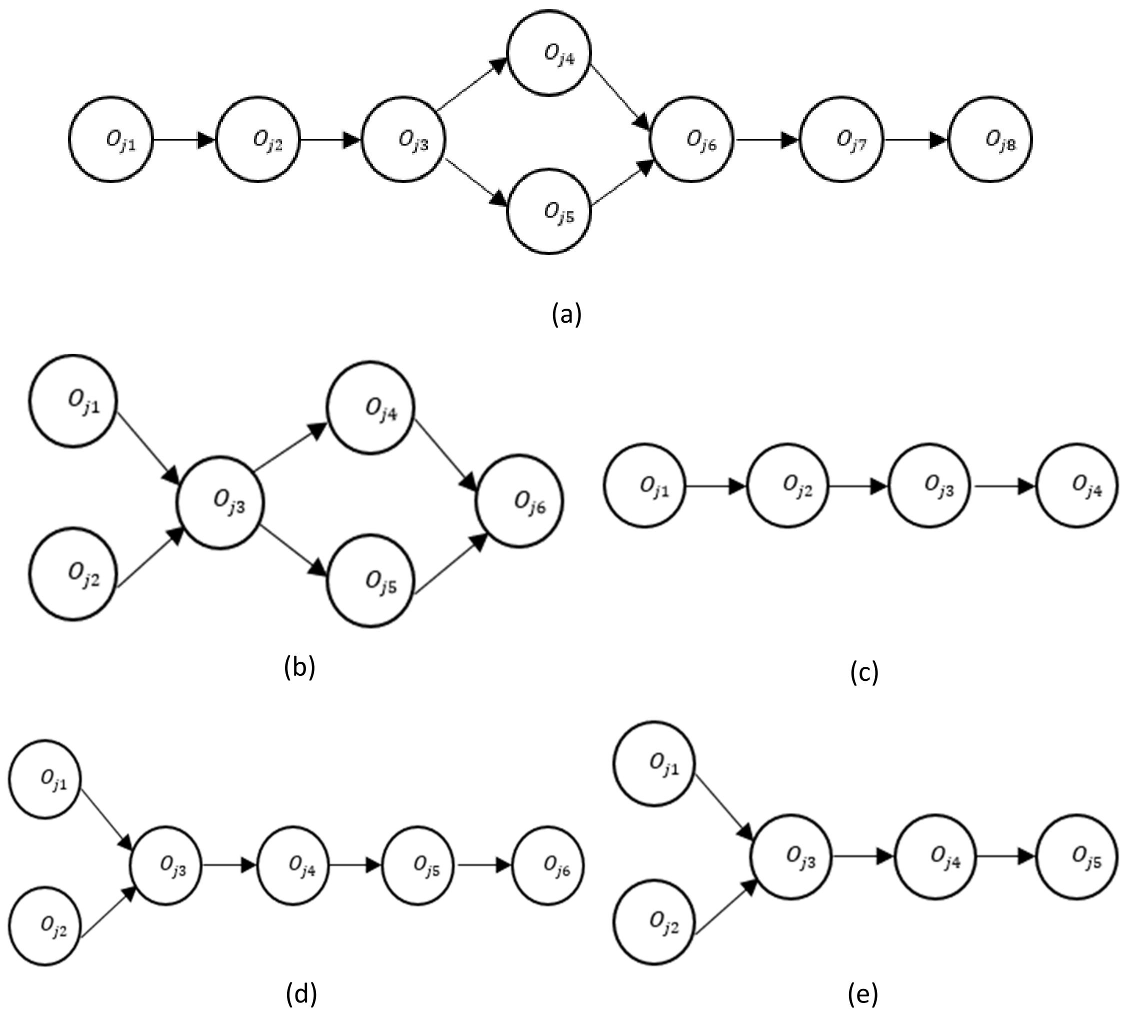



*Figure 6.1 Operation precedence graphs (Case I, II and III).*

*Table 6.1 Case I Operation precedence graph.*

| Job *j* | Operation precedence graph (Figure 6.1) |
|---|---|
| 1,2 and 3 | (a) |
| 4 | (b) |
| 5 | (c) |
| 6, 7, 8 and 10 | (d) |
| 9 | (e) |

Table 6.2 Case II Operation precedence graph.

| Job *j* | Operation precedence graph (Figure 6.1) |
|---|---|
| 1 and 4 | (b) |
| 2 and 5 | (c) |
| 3 | (e) |

Table 6.3 Case III Operation precedence graph.

| Job *j* | Operation precedence graph (Figure 6.1) |
|---|---|
| 1, 2, 3 and 4 | (b) |
| 5 and 8 | (c) |
| 6 and 7 | (e) |

## 6.1 Implementation (a decision support system)

A customized decision support system (DSS) has been developed for an SME manufacturer and vendor of inserts located in the southwest of Canada. Rule-based algorithms and HBFOA has been embedded in the DSS. The DSS has been implemented with Visual Basic for Applications (VBA) on the top of MS Excel.

The DSS is presented by different displays on the excel spreadsheet that contain command buttons, option buttons, check boxes and instructions boxes. The user interacts with the system by inputting the order information, selecting the solution method, reviewing the results and creating a Gantt chart. The DSS use case diagram is presented in Figure 6.2.

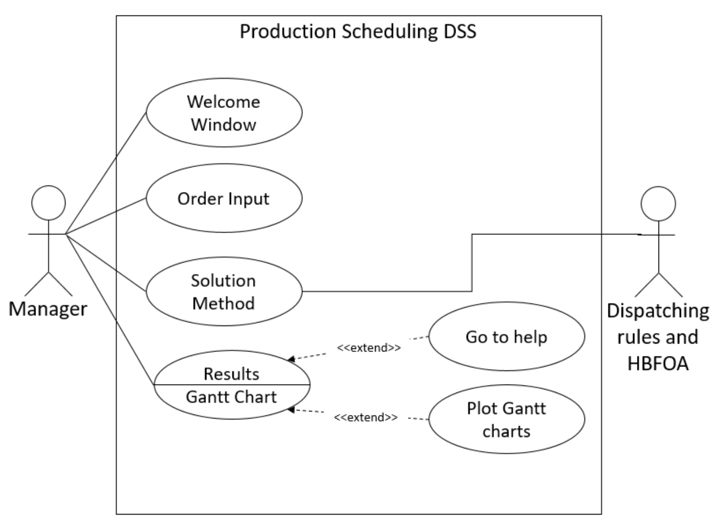



*Figure 6.2 DSS Use case diagram.*

A structure diagram is shown in Figure 6.3. The diagram shows the breakdown of the system that is comprised of seven modules. The DSS starts by inputting jobs information (part numbers). Then the system validates the date and calculates the processing times. Consequently, the solution method is activated. In this stage, either the dispatching rules module or the HBFOA is activated. These modules return scheduling information. After that, the scheduling information is passed to the output schedule module where the schedule is printed. Finally, the information is transferred to plot Gantt chart module to print the required chart.

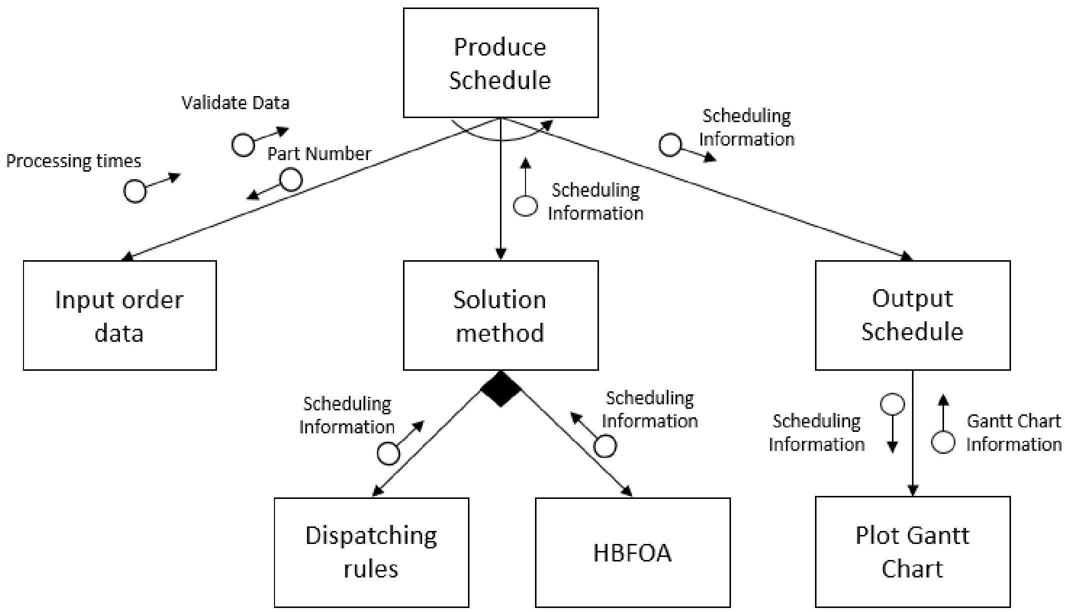


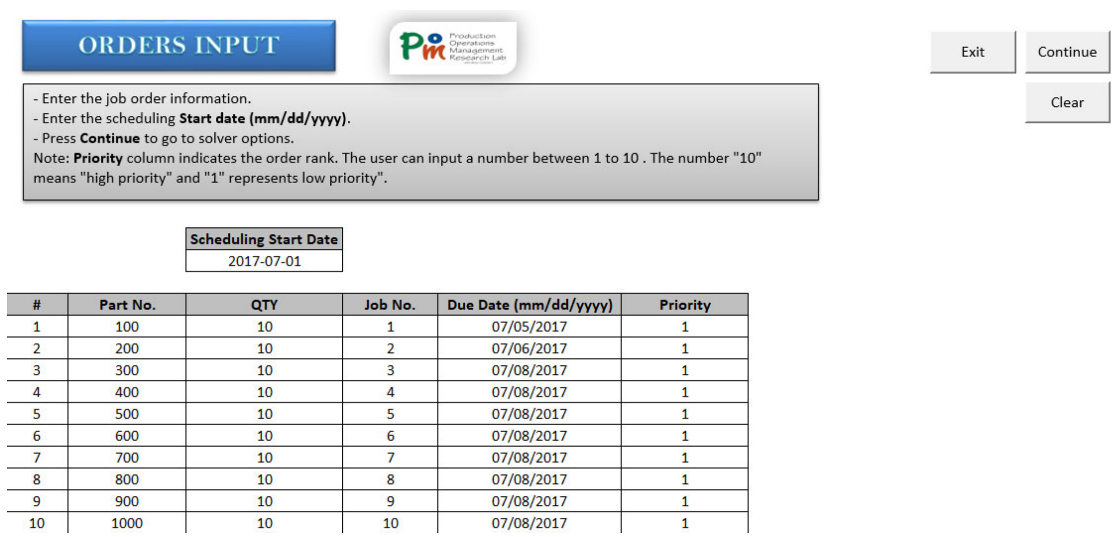

*Figure 6.3 DSS Structure chart.*

### 6.1.1 Spreadsheets overview

The DSS initiates with a *welcome display* which is shown in Figure 6.4. The *welcome display* provides a brief description of the DSS. In this display, the user can either Run Demo or Start inputting jobs. The *Run Demo* button gives a chance to the user to explore the DSS by inputting a set of jobs data that will be required for the DSS functionality. The start button sends the user to the *order input display*.


*Figure 6.4 DSS Welcome display.*

The second display is entitled *order input display,* and it is shown in Figure 6.5. It presents a spreadsheet where the user can input the required job information as part number, quantity, job number, priority and due date. The priority is the weight $W_j$ of the job. Three command buttons are shown in this display (exit, continue and clear).




| # | Part No. | QTY | Job No. | Due Date (mm/dd/yyyy) | Priority |
|---|---|---|---|---|---|
| 1 | 100 | 10 | 1 | 07/05/2017 | 1 |
| 2 | 200 | 10 | 2 | 07/06/2017 | 1 |
| 3 | 300 | 10 | 3 | 07/08/2017 | 1 |
| 4 | 400 | 10 | 4 | 07/08/2017 | 1 |
| 5 | 500 | 10 | 5 | 07/08/2017 | 1 |
| 6 | 600 | 10 | 6 | 07/08/2017 | 1 |
| 7 | 700 | 10 | 7 | 07/08/2017 | 1 |
| 8 | 800 | 10 | 8 | 07/08/2017 | 1 |
| 9 | 900 | 10 | 9 | 07/08/2017 | 1 |
| 10 | 1000 | 10 | 10 | 07/08/2017 | 1 |

*Figure 6.5 DSS Order input window.*

After the user has inputted the job data, the DSS takes the user to the *solution method display* which is shown in Figure 6.6. A brief explanation of the available solutions methods is presented. The solutions methods are listed with option buttons, and the user can select a rule-based approach or the HBFOA solution method. In the same manner, the objectives to be minimized are listed with option buttons. The desired objective (Makespan or weighted tardiness) can be selected through this display.

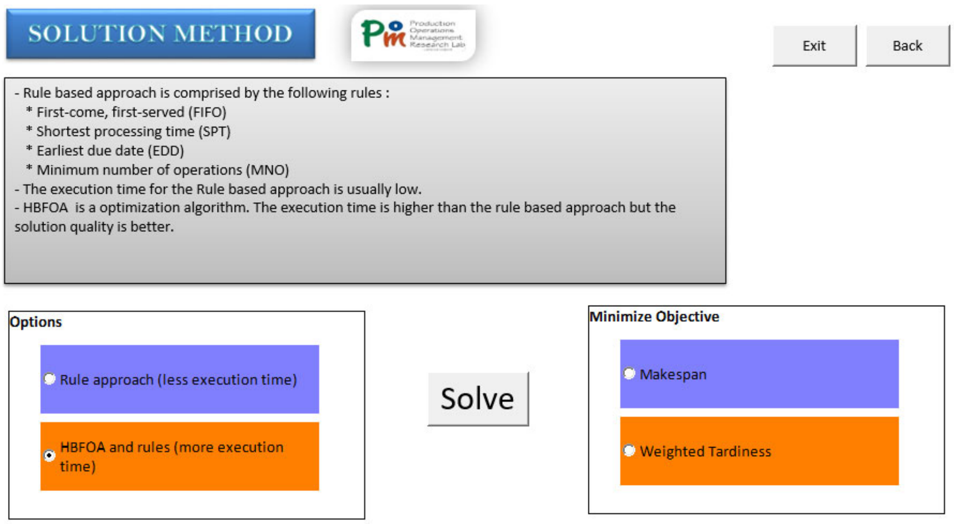



*Figure 6.6 DSS Solution method window.*

After this stage, the DSS will run the solution method desired, and the *results display* is presented. A screenshot for the *Results display* is shown in Figure 6.7. This display contains a section where essential definitions and instructions about the results. A summary is delivered with information regarding the best solution found by the DSS. The information is sorted based on the objective selected in the previous display. The weighted tardiness is presented under the *Score* column. Expanded result tables are provided where additional information as start date, tardiness, shipping date, and the due date is offered. The DSS provides an option to export the *Result display* by pressing the *Export* command button. The *Result display* is provided as a new file that the user can save it for further reference. The user has the option to plot a Gantt chart for the different solutions found. The user selects the required Gantt charts, and after pressing the command button *Plot Gantt Chart,* the charts are displayed in the *Gantt Chart Display.*

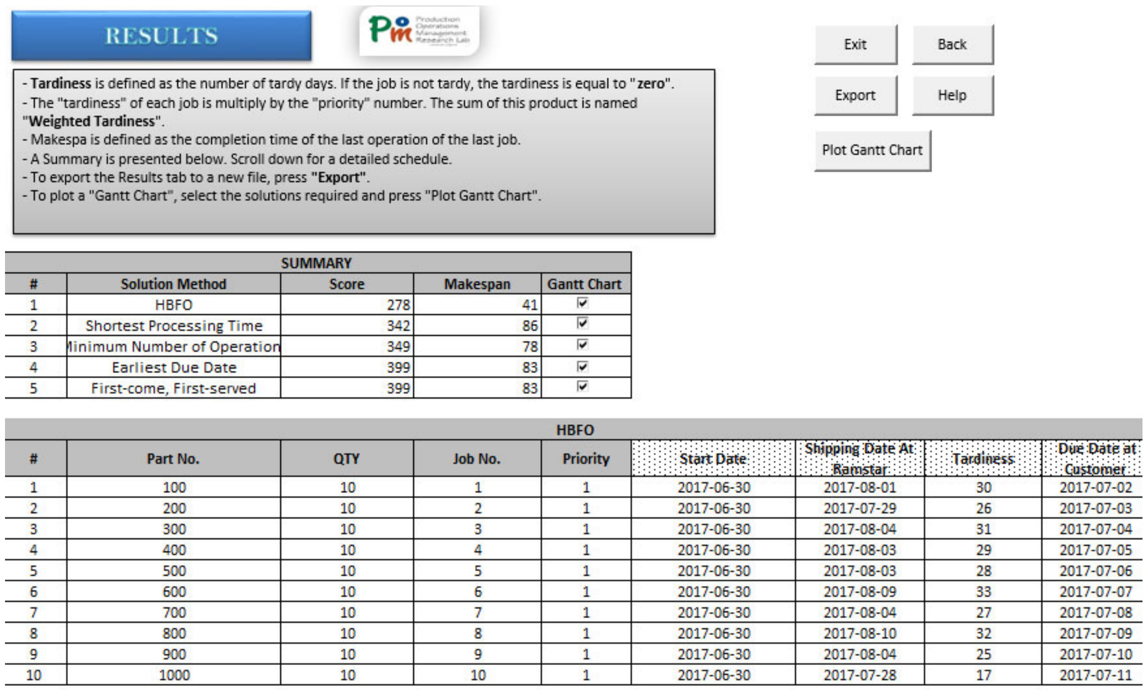



*Figure 6.7. DSS Results window.*

The *Gantt Chart Display* is shown in Figure 6.8. This display presents the set of jobs with a different colour to identify them easily. The Gantt chart can be exported as a new file by pressing the command button *Export*. Finally, the *Results display* provides a *help* command button where the user is transferred to a display where an example of the weighted tardiness calculation is provided. The help display is shown in Figure 6.9.

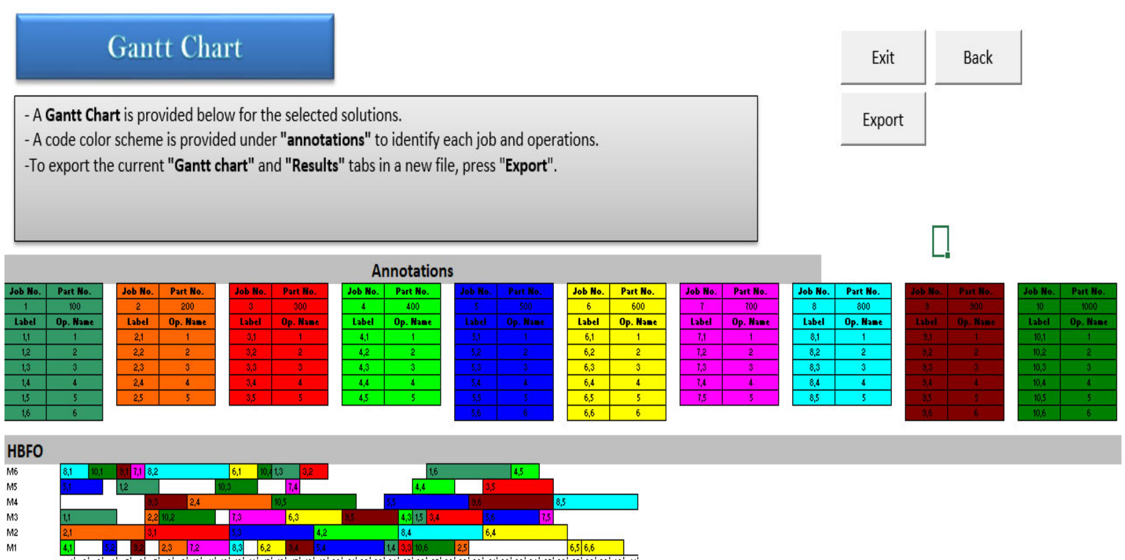



*Figure 6.8 Gantt chart window.*

Help

Back

- The **"Score"** is used to measure the solution quality. The **"Score"** is calculated using the weighted tardiness (WT) as objective. An example is provided in the table below.
- **Tardiness** is defined as the number of tardy days (Shipping date – Due date). If the job is not tardy (zero or negative value), the tardiness is equal to "**zero**". The tardiness of a certain solution can be found in **"column F"**.
- The **"tardiness"** of each job is multiply by the **"priority"** number. This operation can be found in **"column H"**
- The sum of this product is named **"Weighted Tardiness"** and in this system is shown as Score in cell **"H31"**.
- The weighted tardiness works as the objective to be optimized. As a result, the weighted tardiness is minimized.

**Score Calculation example**

| A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|
| # | Part No. | QTY | Shipping Date At Ramstar | Due Date at Customer | Tardiness (D - E) | Priority | F * G |
| 1 | 100 | 10 | 2017-08-17 | 2017-08-18 | 0 | 1 | 0 |
| 2 | 200 | 10 | 2017-09-09 | 2017-08-07 | 33 | 1 | 33 |
| 3 | 300 | 10 | 2017-10-20 | 2017-09-12 | 38 | 1 | 38 |
| 4 | 400 | 10 | 2017-11-12 | 2017-09-06 | 67 | 1 | 67 |
| 5 | 500 | 10 | 2017-12-06 | 2017-10-06 | 61 | 1 | 61 |
| 6 | 600 | 10 | 2017-10-28 | 2017-09-07 | 51 | 1 | 51 |
| 7 | 700 | 10 | 2017-11-30 | 2017-08-31 | 91 | 1 | 91 |
| 8 | 800 | 10 | 2017-11-27 | 2017-12-05 | 0 | 1 | 0 |
| 9 | 900 | 10 | 2017-12-14 | 2017-11-25 | 19 | 1 | 19 |
| 10 | 1000 | 10 | 2017-12-23 | 2017-11-26 | 27 | 1 | 27 |
| | | | | | | Score (Sum of column H) | 387 |

*Figure 6.9 Help display.*

## 6.2 Summary

This chapter presents a case study from a carbide tool and inserts vendor. The company is a typical example of an FJSP where sequencing flexibility is shown. The company has experienced an increased product mix. New and more advanced manufacturing systems have been brought to the manufacturer. Data regarding processing times and precedence relationships has been collected. A DSS has been developed to tackle the production scheduling problem in this company. The HBFOA and rule-based algorithms have been embedded in the system. The DSS has been implemented on VBA and packaged on the top of MS Excel.

# CHAPTER 7. NUMERICAL EXPERIMENTS

In this chapter, instances from the literature and new generated are used to show the applicability and efficiency of the two methodologies proposed for the FSJP. The results of the introduced methods are compared with the literature results.  This chapter is divided into four subsections. The first section presents the procedure to perform the design of experiments in order to calibrate the parameters of the proposed metaheuristics. The second section introduces the instances and results of the application of the FJSPS. Likewise, the third section presents the instances and results of the application of the DRFJSPS. Finally, the last section provides the implementation of the FJSP-2F model in several instances.

## 7.1 Design of experiments

It is important to fine tune the required parameters, to execute both metaheuristics (HBFOA and NSGA-II) successfully. Some authors only set the parameter value based on their experience (Li and Gao 2016; Xia and Wu 2005; Kaplanoğlu 2016) and no further analysis is performed. Other authors performed experiments by inputting different parameter values to try to find a combination that reached the best results on average (González *et al.* 2015). These methodologies do not represent a proper analysis to get the significant factors that are responsible for the algorithm performance.  Design of experiments (DOE) has been used as a technique to calibrate the parameters. Some authors have employed the Taguchi method (Gong *et al.* 2018; Zheng and Wang 2016) to investigate the effects of the parameters with a moderate complexity instance. Similarly, (Vital Soto *et al.* 2017) developed a methodology that involves the use of  a $2^k$ factorial design and regression to calibrate a evolutionary algorithm. This last approach is the one developed in this study and the methodology is described in this chapter.

Due to a large number of parameters, a full factorial design will require a high number of runs. To overcome this situation, a $2^k$ factorial design can be performed. This kind of experiments are employed to analyze screening experiments at early stages. This design works by setting only two levels of each factor. A high level and a low level per factor. The success of this kind of experiments are based on that when there are many factors involved in a system, most probably main effects and low-order interactions are primarily responsible of the effect on the system response (Montgomery 2008).  The $2^k$ designed is employed to detect the significant factors. Then, factors that are not significant are removed from the experiment. After, that the regression equation is used to calculate the expected response. The combination that provides the best yield is the one selected to be used in the numerical experiments. Further analysis could be done by performing a confirmation test and generate surface plots. The methodology is illustrated in Figure 7.1

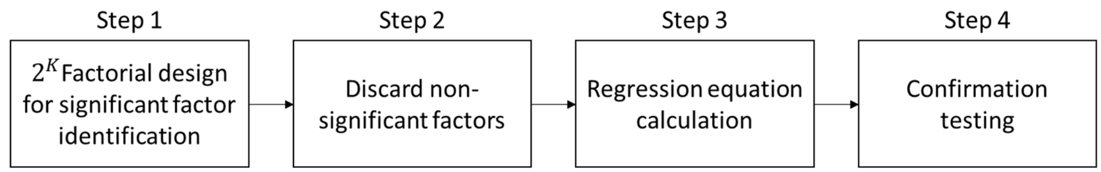



*Figure 7.1DOE methodology.*

### 7.1.1 Design of experiments for the HBFOA

A design of experiment is selected to tune the parameters of the proposed HBFOA. Seven factors have been identified to be part of the design of experiment. In particular a $2^{7-3}$ design with two replicates has been selected with a total of 32 runs. The parameters in the study and their levels are listed as follows:

- Swarm Size ($S$)
- Elimination-Dispersal ($N_{ed}$)
- Reproduction ($N_{re}$)
- Chemotaxis ($N_c$)
- Swims ($N_s$)
- Probabilities
  - Elimination- dispersal ($p_{ed}$)
  - Reproduction ($p_{re}$)
  - Swarming ($p_{swa}$)
- Local search ($LS_1|LS_2$)

The factors related to the probabilities of $p_{ed}$,$p_{re}$ and $p_{swa}$ has embedded in a single factor (attribute) to reduce the number of runs required. The same case is used for the maximum number of iteration for Local search ($LS_1|LS_2$).

This study considers two different objective functions (Weighted tardiness and makespan). Instances with the highest flexibility have been selected to perform this experiment. In the case of the weighted tardiness the instance “Case III” has been used for the experiment. The instance “MK02” instance provide by Brandimarte (1993) have been employed for the experiment with makespan as the objective function. Table 7.1 shows the levels of each parameter in the experiment. An analysis of variance (ANOVA) for the $2^{7-3}$ design has been executed using the statistical software Minitab 17 to find the effect of the parameters in the two different objectives.

*Table 7.1 Parameter levels for the HBFOA.*

| Parameters | Weighted tardiness | | Makespan | |
|---|---|---|---|---|
| | High | Low | High | Low |
| Swarm Size ($S$) | 80 | 40 | 80 | 40 |
| Elimination-Dispersal ($N_{ed}$) | 6 | 3 | 6 | 3 |
| Reproduction ($N_{re}$) | 4 | 2 | 4 | 2 |
| Chemotaxis ($N_c$) | 4 | 2 | 4 | 2 |
| Swims ($N_s$) | 120 | 60 | 120 | 60 |
| Probability ($p_{ed}\|p_{re}\|p_{swa}$) | 0.3 | 0.15 | 0.4 | 0.15 |
| Local search ($LS_1\|LS_2$) | 4500\|1000 | 400\|100 | 6000\|3000 | 400\|100 |

The ANOVA results for the makespan can be shown from Table 7.2. It can be observed that the main effects, namely Swarm Size ($S$), Elimination-Dispersal ($N_{ed}$), Reproduction ($N_{re}$), Local search ($LS_1|LS_2$), and two-way interaction $S * N_{re}$ have $p - values\ < 0.05$. Therefore, these are the significant factors as per Table 7.3. The non-significant factors are removed from the experiment and then, the expected response (makespan) is estimated using the regression Eq. (7.1). The non-significant factors are set to a low level to decrease the computational efforts. The minimum response can be found when $S = 40$, $N_{ed} = 4$, $N_{re} = 4$, $N_c = 2$, $N_s = 60$, $p_{ed}|p_{re}|p_{swa} = 0.15$ and $LS_1|LS_2 = (6000|3000)$. This setting will be used to conduct the numerical experiments.

$$\hat{y} = 27.937 - 0.313(S) - 0.437(N_{ed}) - 0.312(N_{re}) - 0.625(LS_1|LS_2) + 0.313\,(S) * (N_{ed}) \tag{7.1}$$

*Table 7.2 Analysis of variance output from Minitab for the makespan.*

| Source | DF | Adj SS | Adj MS | F-Value | P-Value |
|---|---|---|---|---|---|
| Model | 15 | 33.8750 | 2.2583 | 4.52 | 0.002 |
| Linear | 7 | 26.0000 | 3.7143 | 7.43 | 0.000 |
| $S$ | 1 | 3.1250 | 3.1250 | 6.25 | 0.024 |
| $N_{ed}$ | 1 | 6.1250 | 6.1250 | 12.25 | 0.003 |
| $N_{re}$ | 1 | 3.1250 | 3.1250 | 6.25 | 0.024 |
| $N_c$ | 1 | 0.5000 | 0.5000 | 1.00 | 0.332 |
| $N_s$ | 1 | 0.1250 | 0.1250 | 0.25 | 0.624 |
| $p_{ed}/p_{re}/p_{swa}$ | 1 | 0.5000 | 0.5000 | 1.00 | 0.332 |
| $LS_1/LS_2$ | 1 | 12.5000 | 12.5000 | 25.00 | 0.000 |
| 2-Way Interactions | 7 | 7.8750 | 1.1250 | 2.25 | 0.085 |
| $S * N_{ed}$ | 1 | 3.1250 | 3.1250 | 6.25 | 0.024 |
| $S * N_{re}$ | 1 | 1.1250 | 1.1250 | 2.25 | 0.153 |
| $S * N_c$ | 1 | 0.0000 | 0.0000 | 0.00 | 1.000 |
| $S * N_s$ | 1 | 1.1250 | 1.1250 | 2.25 | 0.153 |
| $S * p_{ed}/p_{re}/p_{swa}$ | 1 | 0.0000 | 0.0000 | 0.00 | 1.000 |
| $S * LS_1/LS_2$ | 1 | 2.0000 | 2.0000 | 4.00 | 0.063 |
| $N_{ed} * N_c$ | 1 | 0.5000 | 0.5000 | 1.00 | 0.332 |
| 3-Way Interactions | 1 | 0.0000 | 0.0000 | 0.00 | 1.000 |
| $S * N_{ed} * N_c$ | 1 | 0.0000 | 0.0000 | 0.00 | 1.000 |
| Error | 16 | 8.0000 | 0.5000 | | |
| Total | 31 | 41.8750 | | | |
| | | | | | |
| Model Summary | | | | | |
| S | R-sq | R-sq(adj) | R-sq(pred) | | |
| 0.707107 | 80.9% | 62.99% | 23.58% | | |

*Table 7.3 Significant factors of HBFOA for the makespan.*

| Parameters |
|---|
| Swarm Size ($S$) |
| Elimination-Dispersal ($N_{ed}$) |
| Reproduction ($N_{re}$) |
| Local search ($LS_1 \| LS_2$) |
| 2- Way interaction $S * N_{re}$ |

For the case of the weighted tardiness, the ANOVA results is presented in Table 7.4. The main effect Elimination-Dispersal ($N_{ed}$), two-way interaction $S * N_s$ have $p - values\ < 0.05$ and these are the significant, as per Table 7.5. In spite of the main effect factors Swarm Size ($S$) and Swims ($N_s$) are non significant, the factors needs to be included in the analysis, due to the two-way interaction $S * N_s$ is significant and it presents a weak heredity (Montgomery 2008). The regression equation is estimated to calculate the expected response (weighted tardiness) using Eq. (7.2). The non-significant factors are set it to low to reduce the computational efforts and the minimum response can be found at $S = 40$, $N_{ed} = 5$, $N_{re} = 2$, $N_c = 2$, $N_s = 120$, $p_{ed}|p_{re}|p_{swa} = 0.15$ and $LS_1|LS_2 = (400/100)$. This parameter configuration is used to conduct the numerical experiments.

$$\hat{y} = 129.81 - 1.37(S) - 6.31(N_{ed}) - 1.75(N_s) + 5.56\ (S) * (N_s) \tag{7.2}$$

*Table 7.4 Analysis of variance output from Minitab for the weighted tardiness.*

| Source | DF | Adj SS | Adj MS | F-Value | P-Value |
|---|---|---|---|---|---|
| Model | 15 | 4061.87 | 270.79 | 1.29 | 0.309 |
| Linear | 7 | 2059.88 | 294.27 | 1.40 | 0.271 |
| $S$ | 1 | 60.50 | 60.50 | 0.29 | 0.599 |
| $N_{ed}$ | 1 | 1275.12 | 1275.12 | 6.08 | 0.025 |
| $N_{re}$ | 1 | 595.13 | 595.13 | 2.84 | 0.112 |
| $N_c$ | 1 | 2.00 | 2.00 | 0.01 | 0.923 |
| $N_s$ | 1 | 98.00 | 98.00 | 0.47 | 0.504 |
| $p_{ed}/p_{re}/p_{swa}$ | 1 | 8.00 | 8.00 | 0.04 | 0.848 |
| $LS_1/LS_2$ | 1 | 21.13 | 21.13 | 0.10 | 0.755 |
| 2-Way Interactions | 7 | 2000.87 | 285.84 | 1.36 | 0.286 |
| $S * N_{ed}$ | 1 | 12.50 | 12.50 | 0.06 | 0.810 |
| $S * N_{re}$ | 1 | 338.00 | 338.00 | 1.61 | 0.223 |
| $S * N_c$ | 1 | 0.13 | 0.13 | 0.00 | 0.981 |
| $S * N_s$ | 1 | 990.12 | 990.12 | 4.72 | 0.045 |
| $S * p_{ed}/p_{re}/p_{swa}$ | 1 | 496.12 | 496.12 | 2.36 | 0.144 |
| $S * LS_1/LS_2$ | 1 | 162.00 | 162.00 | 0.77 | 0.393 |
| $N_{ed} * N_c$ | 1 | 2.00 | 2.00 | 0.01 | 0.923 |
| 3-Way Interactions | 1 | 1.12 | 1.12 | 0.01 | 0.943 |
| $S * N_{ed} * N_c$ | 1 | 1.12 | 1.12 | 0.01 | 0.943 |
| Error | 16 | 3357.00 | 209.81 | | |
| Total | 31 | 7418.87 | | | |
| | | | | | |
| Model Summary | | | | | |
| S | R-sq | R-sq(adj) | R-sq(pred) | | |
| 14.4849 | 54.75% | 12.33% | 0.00% | | |

*Table 7.5 Significant factors of HBFOA for the weighted tardiness.*

| Parameters |
|---|
| Elimination-Dispersal ($N_{ed}$) |
| 2- Way interaction $S * N_s$ |

### 7.1.2 Design of experiments for the NSGA-II

The NSGA-II employs three factors: Number of Generations (t), Mutation probability ($M_p$) and Population size (N). Table 7.6 shows the levels of each parameter in the experiment. An analysis of variance (ANOVA) for the $2^3$ design with 2 replicates. A total of 24 runs has been executed. The statistical software Minitab 17 has been used to identify the effect of the parameters. The instance with hithe ghest flexibility (DR1) is used to conduct the experiment. Our response variable is defined as the total number of no-dominated solutions provided.

*Table 7.6 Parameter levels for the NSGA-II.*

| Parameters | Levels | |
|---|---|---|
| | High | Low |
| Generations (t) | 40 | 150 |
| Mutation rate ($M_p$) | 0.1 | 0.5 |
| Population (N) | 20 | 150 |

The ANOVA for the multi-objective problem can be shown from Table 7.7. It can be observed that the main effects Generations ($G$) and Population ($P$) have $p - values\ < 0.05$. Therefore, these are the significant factors as per Table 7.7. The non-significant factor is removed from the experiment and then, the expected response is estimated using the regression Eq. (7.3). The non-significant factor is set to $M_p = 0.5$ to enhance the solution space exploration. The minimum response can be found when $G = 40$ and $P = 150$. A surface plot illustration is generated to confirm the results provided by the regression equation. The factors setting found in this experiment as per Table 7.8 , are used to conduct the numerical experiments.

$$\hat{y} = 3.417 - 1.250\ (G) + 0.917\ \ (P) \tag{7.3}$$

*Table 7.7 Analysis of variance output from Minitab for the multi-objective problem.*

| Source | DF | Adj SS | Adj MS | F-Value | P-Value |
|---|---|---|---|---|---|
| Model | 7 | 105.167 | 15.024 | 4.40 | 0.007 |
| Linear | 3 | 71.167 | 23.722 | 6.94 | 0.003 |
| $t$ | 1 | 37.500 | 37.500 | 10.98 | 0.004 |
| $M_p$ | 1 | 13.500 | 13.500 | 3.95 | 0.064 |
| $N$ | 1 | 20.167 | 20.167 | 5.90 | 0.027 |
| 2-Way Interactions | 3 | 25.833 | 8.611 | 2.52 | 0.095 |
| $t * M_p$ | 1 | 4.167 | 4.167 | 1.22 | 0.286 |
| $t * N$ | 1 | 8.167 | 8.167 | 2.39 | 0.142 |
| $M_p * N$ | 1 | 13.500 | 13.500 | 3.95 | 0.064 |
| 3-Way Interactions | 1 | 8.167 | 8.167 | 2.39 | 0.142 |
| $t * M_p * N$ | 1 | 8.167 | 8.167 | 2.39 | 0.142 |
| Error | 16 | 54.667 | 3.417 | | |
| Total | 23 | 159.833 | | | |
| | | | | | |
| Model Summary | | | | | |
| S | R-sq | R-sq(adj) | R-sq(pred) | | |
| 1.84842 | 65.80% | 50.83% | 23.04% | | |

*Table 7.8 Setting for the significant factors of NSGA-II for the multi-objective problem.*

| Parameters |
|---|
| Generations $(G)$ |
| Population $(P)$ |

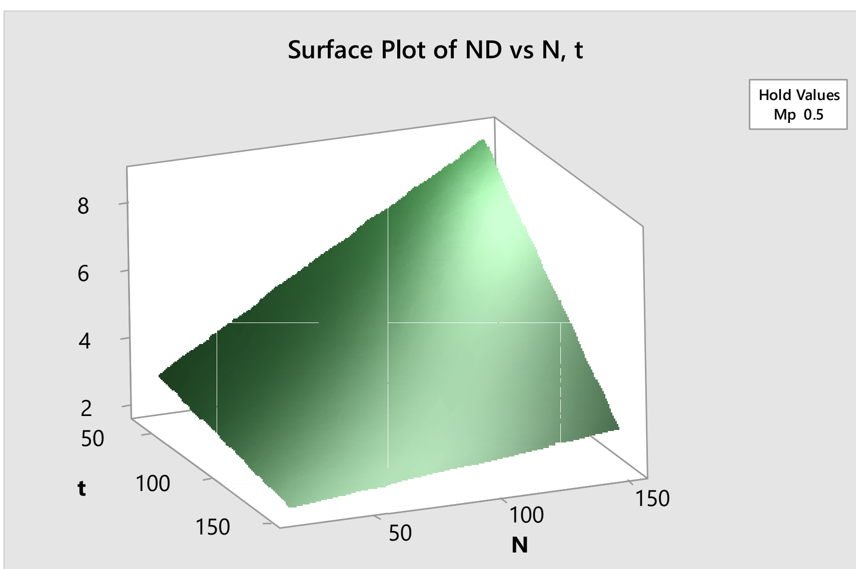



*Figure 7.2 Surface plot of factors ND vs N and t*

## 7.2 Numerical experience for the FJSPS

Xpress Optimizer 25.01.05 algebraic model language and optimizer have been used to code the mathematical model. HBFOA was implemented using Excel with Visual Basic for Applications (VBA). The experiment was performed in PC with a processor Intel i3-6100 2.3GHz with 8GB in RAM. The parameters for HBFOA have been decided by the design of experiments. In order to evaluate the performance of the HBFOA, different instances have been considered.

In the case of the weighted tardiness, instances Case I, II, and III described in section CHAPTER 6 have been considered. For makespan case, the well-known set BRdata provide by Brandimarte (Brandimarte 1993). The set is comprised of 10 test problems.

To evaluate the deviation of the different solutions, the relative deviation criterion is provided in Eq. (7.4) where $Obj(HBFOA)$ is the best solution found by the HBFOA and $Obj(comp)$ is the best solution of the algorithm that is compared with our metaheuristic.

$$dev = [(Obj(comp) - Obj(HBFOA))/Obj(comp)] \times 100\% \tag{7.4}$$

The first data set examined is the one comprised of the case I, II, III for the weighted tardiness objective. The MILP formulation could not find an optimal solution for this data set after running the problem for 3600 seconds. Only the best integer solution is reported. In the same manner, the best solution of HBFOA is compared with the classical scheduling dispatching rules (FCFC, SPT, EDD and MNO) where the assignment is done by *AssigmentRule1.* The comparison is provided in Table 7.9. The first and second column shows the name and size of the problem respectively. The third presents the total number of operation (ope), and the fourth column provides information about the problem flexibility (flex) that is defined as the average number of available machines per operations. The fifth column shows the result of the best solution found after running the algorithm five times. The sixth column presents the average result of the five runs. Columns 7-14 presents the solutions provided by FCFC, SPT, EDD, MNO dispatching rules. The relative deviation is provided too. Columns 15-16 show the best integer solution found by ILP after 3600 seconds.

Table 7.9 The experimental results for the weighted tardiness data.

| Case | *n x m* | ope | flex | HBFOA | | FCFC | | SPT | | EDD | | MNO | | MILP | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | WT | AV(WT) | WT | dev | WT | dev | WT | dev | WT | dev | WT | dev |
| I | 10 x 15 | 63 | 2.30 | 225 | 230.6 | 697 | +67.7 | 621 | +63.8 | 530 | +57.5 | 547 | +58.9 | 462 | +51.3 |
| II | 5 x 12 | 25 | 2.40 | 29 | 31.2 | 171 | +83 | 197 | +85.3 | 213 | +86.4 | 248 | +88.3 | 35 | +17.1 |
| II | 8 x 13 | 44 | 2.57 | 109 | 112.6 | 810 | +86.5 | 437 | +75.1 | 551 | +80.2 | 610 | +82.1 | 274 | +60.2 |

The results presented in Table 7.9 reveals that the HBFOA outperformed all the classic dispatching rules and the integer solution by the MILP. For instance, Case I shows a deviation of a minimum of +57.5% for EDD. In the same manner, the HBFOA reports a deviation of +83% for Case II against FCFC. Case III outperformed +75.1% for the SPT rule. The HBFOA improved the solution presented by the MILP by +51.3%, +17.1% and 60.2% for Case I, II and II respectively. Gant charts for the best solution of the HBFOA for Case I, II and III are presented in Figure 7.6, Figure 7.7 and Figure 7.8 respectively. In the same manner bar charts are illustrated in Figure 7.3, Figure 7.4 and Figure 7.5 for the case I, II and II respectively.

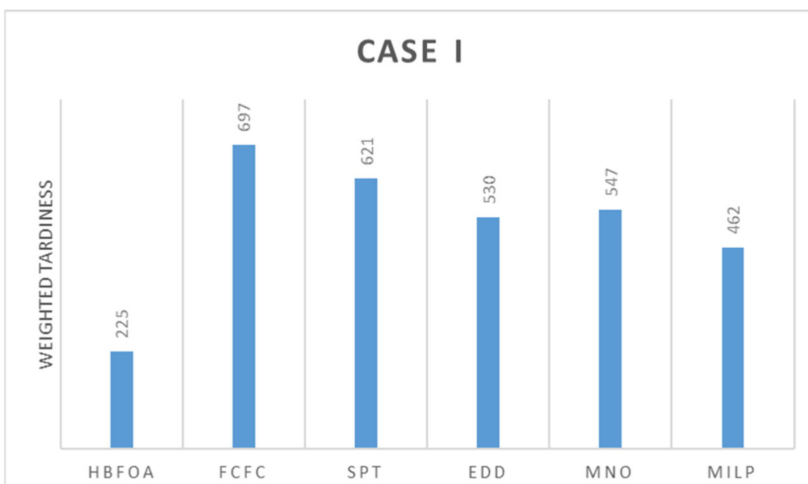



*Figure 7.3 Gantt chart of machines of DR1-solution 6.*

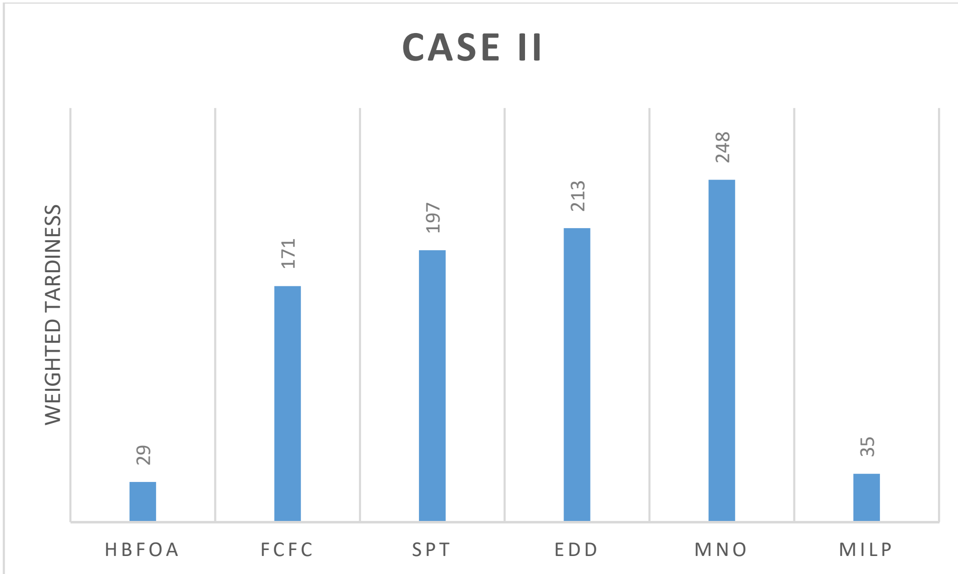



*Figure 7.4 Gantt chart of machines of DR1-solution 6.*

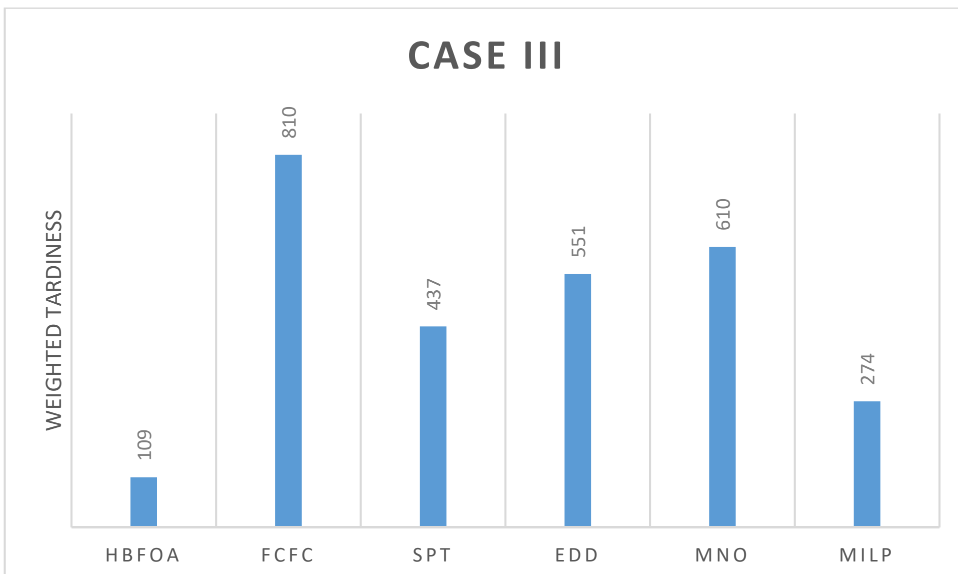



*Figure 7.5 Gantt chart of machines of DR1-solution 6.*

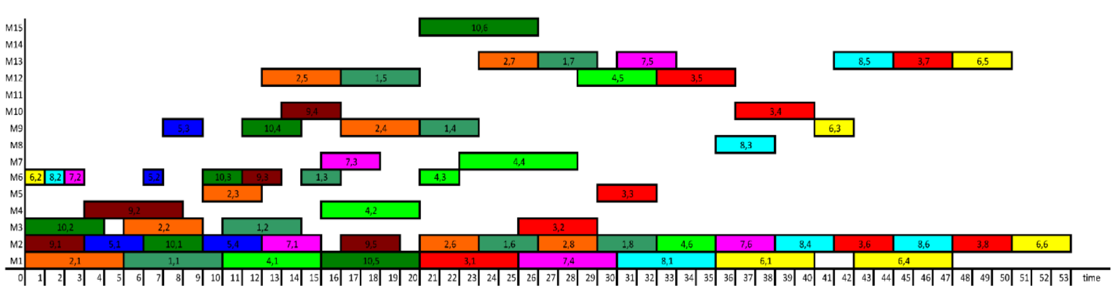



*Figure 7.6 Gantt chart of Case I.*

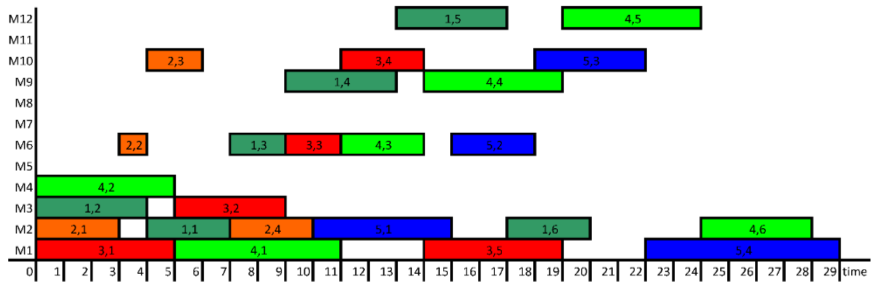



*Figure 7.7 Gantt chart of Case II.*

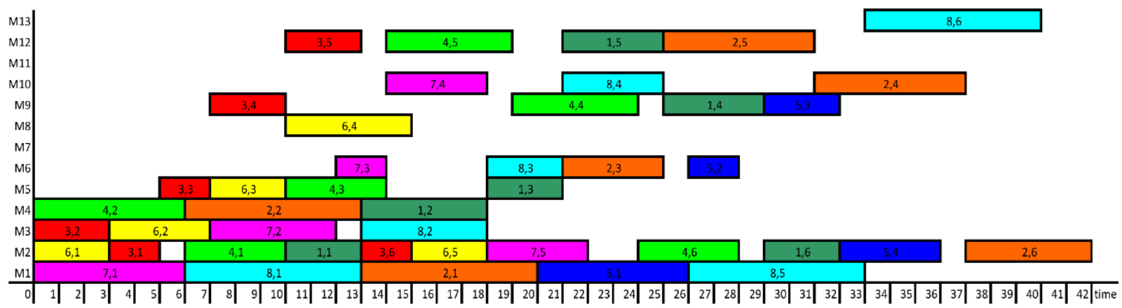


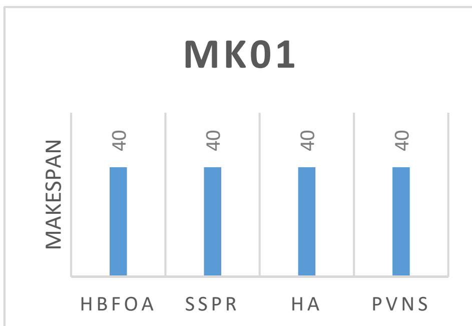

*Figure 7.8 Gantt chart for Case III.*

The results of the second data set (BRdata) for the makespan ($C_{max}$) is revealed in Table 7.10. The HBFOA is compared with three remarkable works in the literature: SSPR of Gonzalez *et al.* (González *et al.* 2015), HA of Li and Gao (Li and Gao 2016) and PVNS of Yazdani *et al.* (Yazdani *et al.* 2010). The first column (ins) shows the problem name. Column 3, 4 and 5 show the problem size, the number of operations and the problem flexibility respectively. Column 5 provides the result of the best makespan found after running the HBFOA five times. The sixth columns present the average result of the five runs. Columns 7, 9, 11 provides the results obtained by SSPR, HA and PVNS. The deviation relative deviation is shown in columns 8, 10 and 12.

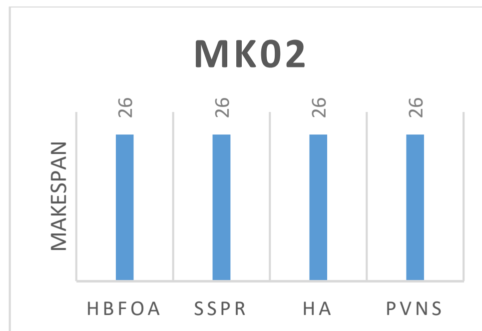
*Table 7.10 The experimental results for the BRdata.*

| ins | *n x m* | ope | flex | HBFOA | | SSPR | | HA | | PVNS | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | $C_{max}$ | AV($C_{max}$) | $C_{max}$ | dev | $C_{max}$ | dev | $C_{max}$ | dev |
| MK01 | 10 x 6 | 55 | 2.09 | 40 | 40 | 40 | 0 | 40 | 0 | 40 | 0 |
| MK02 | 10 x 6 | 58 | 4.10 | 26 | 26.2 | 26 | 0 | 26 | 0 | 26 | 0 |
| MK03 | 15 x 8 | 150 | 3.01 | 204 | 204 | 204 | 0 | 204 | 0 | 204 | 0 |
| MK04 | 15 x 8 | 90 | 1.91 | 60 | 60.4 | 60 | 0 | 60 | 0 | 60 | 0 |
| MK05 | 15 x 4 | 106 | 1.71 | 172 | 173.6 | 172 | 0 | 172 | 0 | 173 | +0.6 |
| MK06 | 10 x 15 | 150 | 3.27 | 57 | 58.8 | 57 | 0 | 57 | 0 | 60 | +5 |
| MK07 | 20 x 5 | 100 | 2.83 | 139 | 140.6 | 139 | 0 | 139 | 0 | 141 | +1.4 |
| MK08 | 20 x 10 | 225 | 1.43 | 523 | 523 | 523 | 0 | 523 | 0 | 523 | 0 |
| MK09 | 20 x 10 | 240 | 2.53 | 307 | 307 | 307 | 0 | 307 | 0 | 307 | 0 |
| MK10 | 20 x 15 | 240 | 2.98 | 205 | 208.6 | 196 | -4.6 | 197 | -4.1 | 208 | +1.4 |

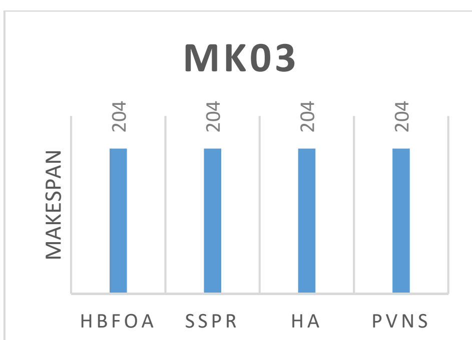
As shown in Table 7.10, the HBFOA outperformed the results provided by PVNS. Relative deviation of +0.6, +5, +1.4 and +1.4 are shown for problems MK05, MK06, MK07 and MK10, respectively. The HBFOA produced equivalent results for the majority of the solutions of SSPR and HA. The HBFOA matches the best solutions known for problems MK01-09. For problem MK10 the relative deviation is -4.6% and -4.1%, respectively compared with SSPR and HA. The HBFOA offered comparable results to the best solutions of SSPR, HA and PVNS. The Gantt charts for problems MK01-10 are provided in Figure 7.11, Figure 7.12, Figure 7.13 and Figure 7.15. Similarly, bar charts are provided in Figure 7.9 and Figure 7.10 for the makespan comparison of problems MK01-10.



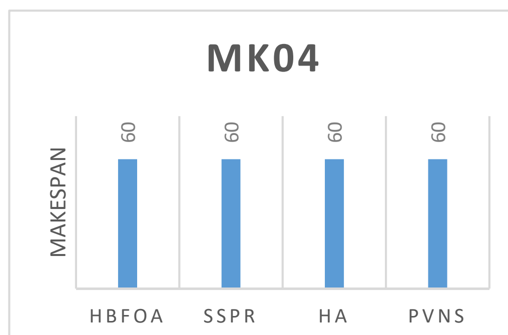

(a)



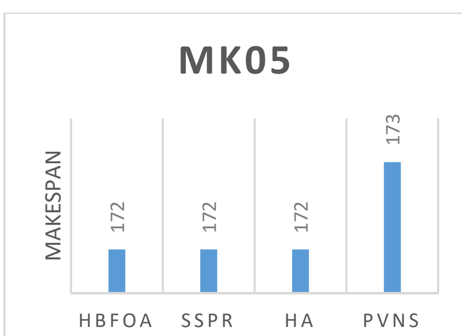

(b)



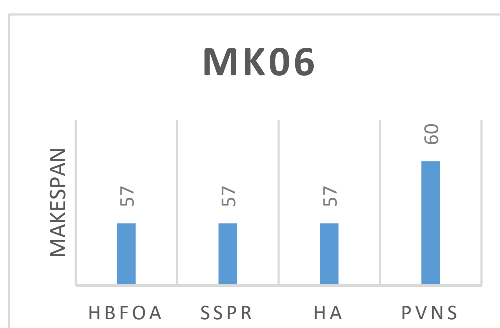

(c)




(d)




(e)




(f)

*Figure 7.9 Gantt chart of  BRdata MK01-Mk06.*

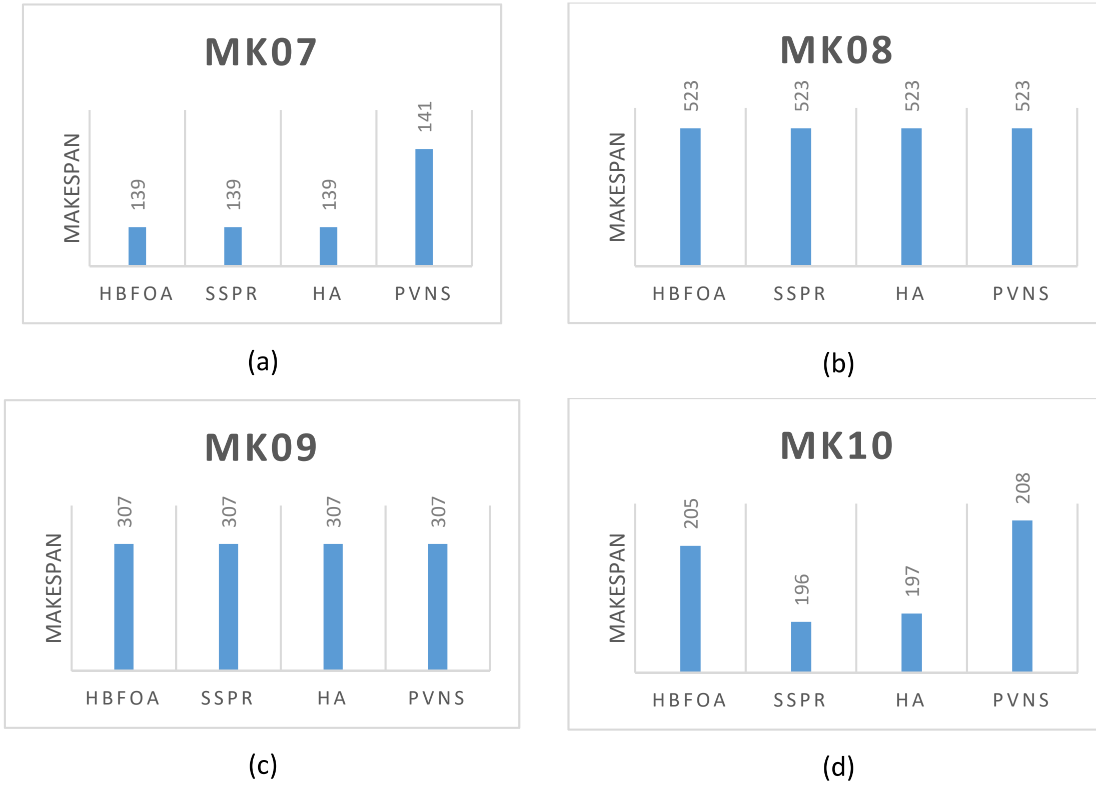



*Figure 7.10 Gantt chart of BRdata MK07-MK10.*

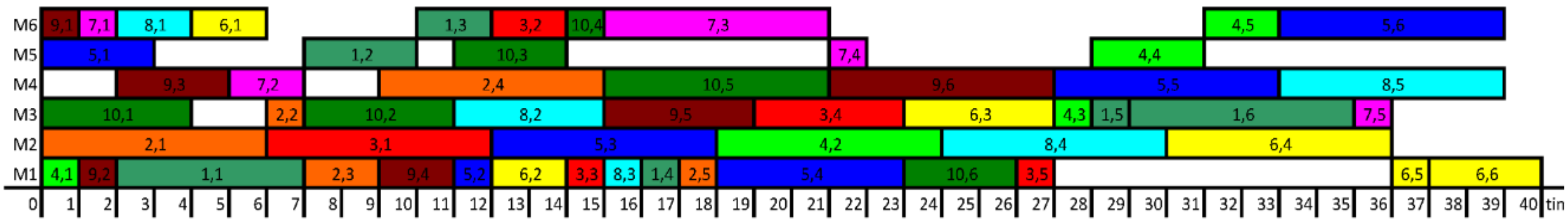



*Figure 7.11 Gantt chart of problem MK01.*

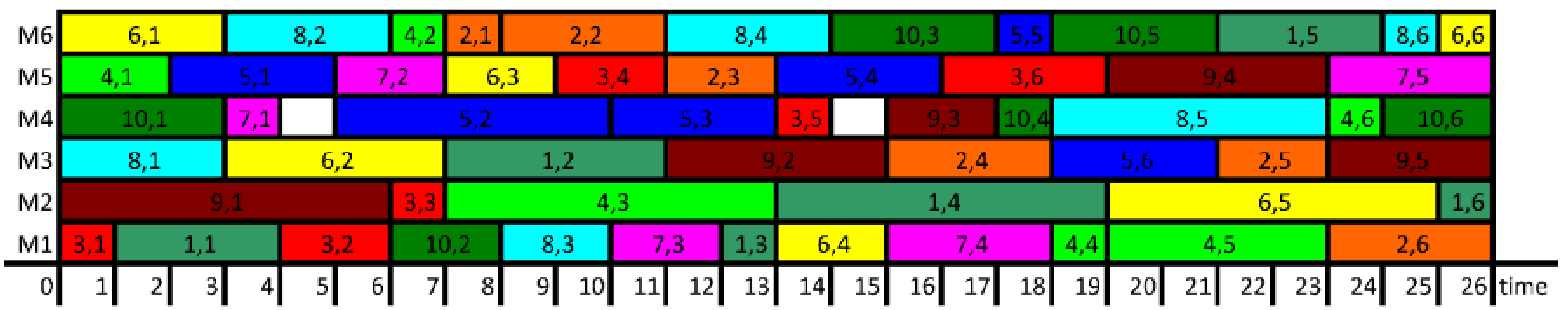



*Figure 7.12 Gantt chart of problem MK02.*

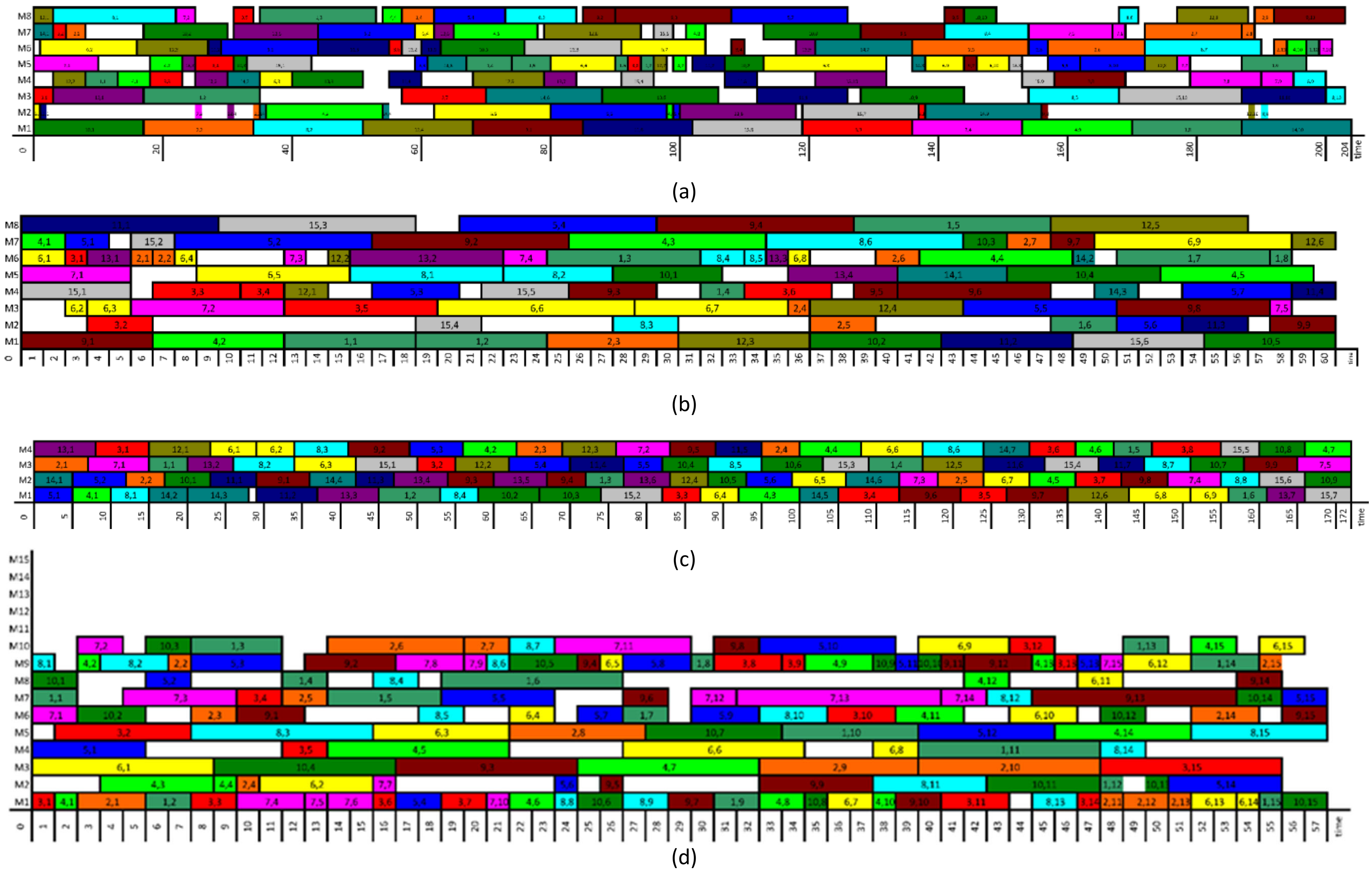
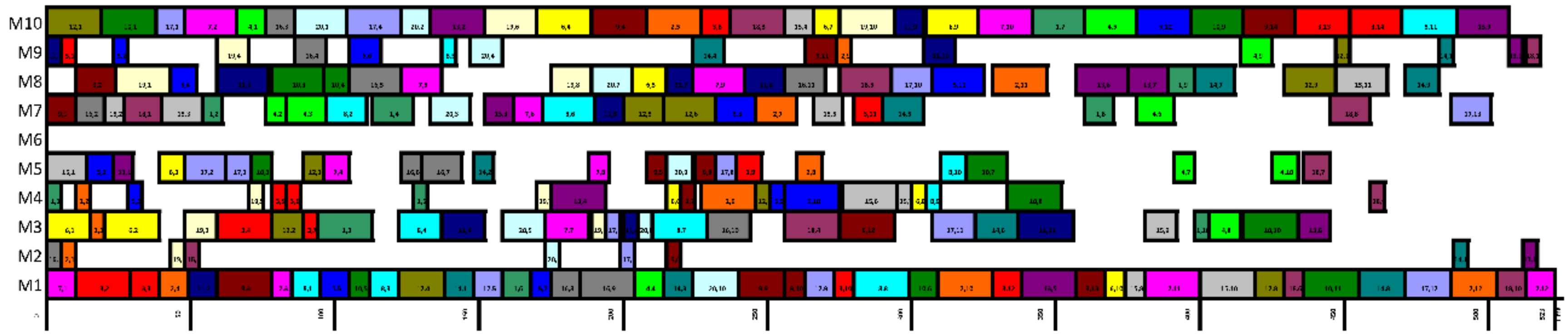
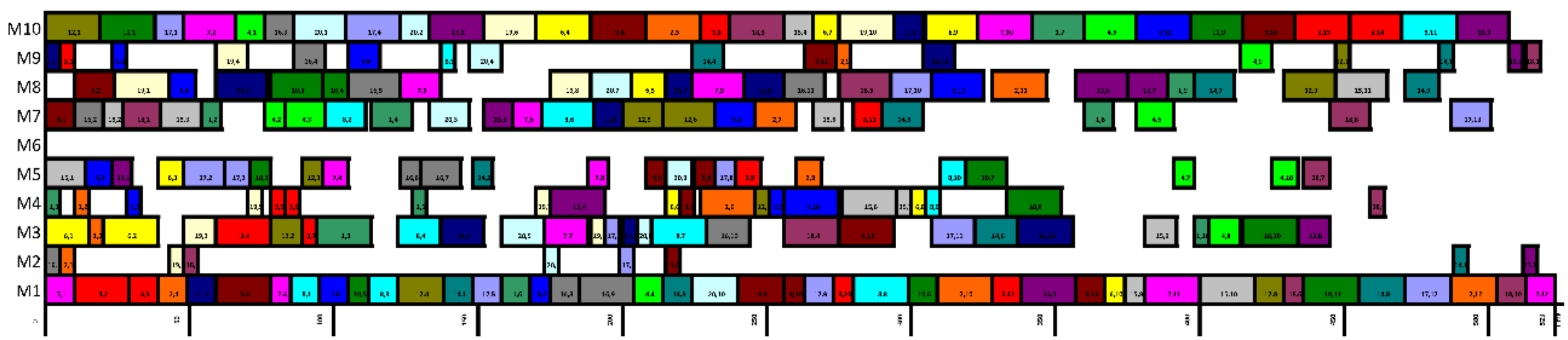


*Figure 7.13 Gantt chart of problem (a) MK03; (b) MK04; (c) MK05; (d)MK06.*

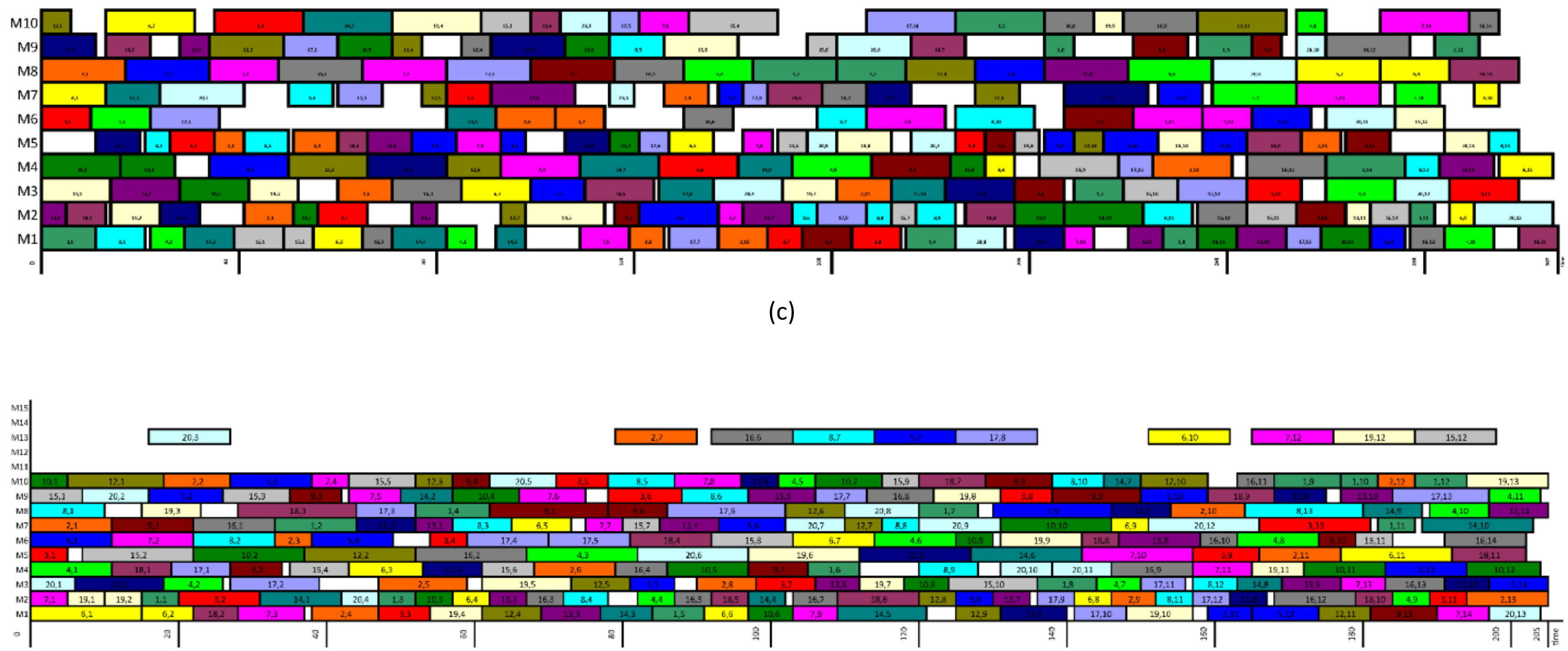



(a)




(b)

*Figure 7.14 Gantt chart of problem (a) MK07 and (b) MK08.*



(c)

(d)

*Figure 7.15 Gantt chart of problem (c) MK09 and (d)MK10.*

## 7.3 Numerical experience for the DRFJSPS

Xpress Optimizer 25.01.05 algebraic model language and optimizer have been used to code the mathematical model and solve the NSGA-II for the DRFJSPS. The experiment was performed in PC with a processor Intel i3-6100 2.3GHz with 8GB in RAM. The parameters for NSGA-II have been decided by the design of experiments. Five instances of different sizes and flexibility have been considered as presented in Table 7.11. Flexibility is defined as the average number of available resources per operation.

Instance DR-1, DR-2 and DR-3 are adaptations of the instances provided by the case study detailed in CHAPTER 6. The remaining instances were generated from a uniform distribution, and the parameter are presented in Table 7.11. The data for the five instances are presented in APPENDIX A (Table A.5 to Table A.28). Operations precedence graphs can be found in Figure 7.16. The data to construct the graphs used for DR1, DR2, DR3, DR4 and DR5 are presented in Table 7.12, Table 7.13, Table 7.14, Table 7.15 and Table 7.16 respectively.

*Table 7.11 Instances for the (DRFJSPS).*

| Instance | Jobs | Machines | Workers | Operations | Flexibility | $P_{iujk}$ | $W_j$ | $d_j$ |
|---|---|---|---|---|---|---|---|---|
| DR 1 | 10 | 15 | 10 | 63 | 85.57 | 5-20 | N/A | N/A |
| DR2 | 5 | 12 | 3 | 25 | 19.76 | 2-10 | N/A | N/A |
| DR 3 | 8 | 13 | 4 | 44 | 23.64 | 3-15 | N/A | N/A |
| DR 4 | 6 | 6 | 3 | 41 | 8.83 | 4-8 | 1-10 | 6-20 |
| DR 5 | 7 | 8 | 5 | 47 | 22.7 | 2-15 | 1-10 | 10-35 |

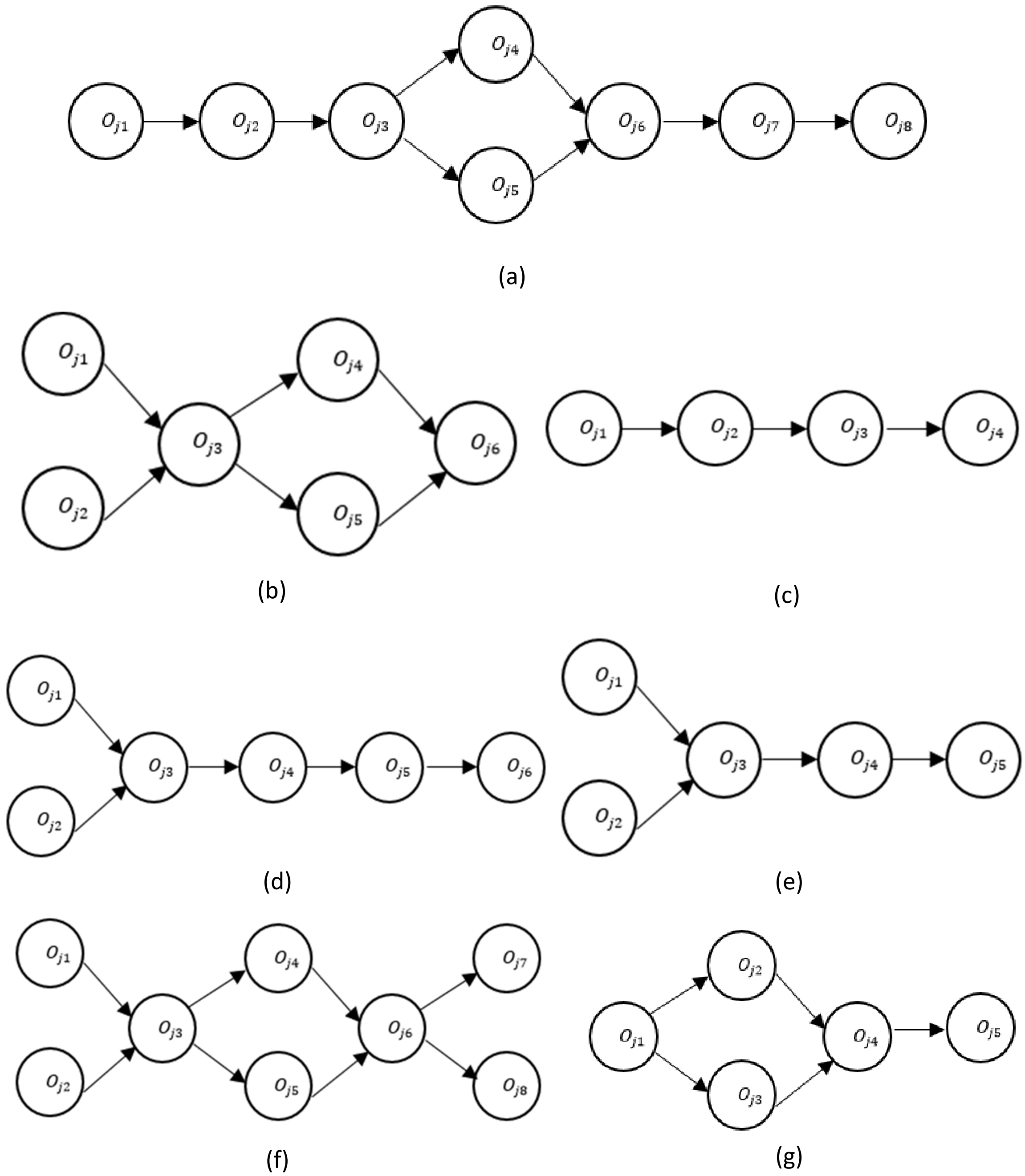



*Figure 7.16 Operation precedence graphs DR1, DR2, DR3, DR4 and DR5.*

*Table 7.12 DR1 operation precedence graph.*

| Job *j* | Operation precedence graph (Figure 7.16) |
|---|---|
| 1,2 and 3 | (a) |
| 4 | (b) |
| 5 | (c) |
| 6, 7, 8 and 10 | (d) |
| 9 | (e) |

*Table 7.13 DR2 operation precedence graph.*

| Job *j* | Operation precedence graph (Figure 7.16) |
|---|---|
| 1 and 4 | (b) |
| 2 and 5 | (c) |
| 3 | (e) |

*Table 7.14 DR3 operation precedence graph.*

| Job *j* | Operation precedence graph (Figure 7.16) |
|---|---|
| 1, 2, 3 and 4 | (b) |
| 5 and 8 | (c) |
| 6 and 7 | (e) |

*Table 7.15 DR4 operation precedence graph.*

| Job *j* | Operation precedence graph (Figure 7.16) |
|---|---|
| 1, 2, and 3 | (f) |
| 4 | (g) |
| 5 and 6 | (b) |

*Table 7.16 DR5 operation precedence graph.*

| Job *j* | Operation precedence graph (Figure 7.16) |
|---|---|
| 1 and 2 | (a) |
| 3 | (e) |
| 4 and 5 | (f) |
| 6 and 7 | (g) |

### 7.3.1 Multi-random-start local search

Multi-random-start local search (MRSLS) is an optimization method is a specifically developed algorithm that is regularly used to assess the performance of metaheuristic. For this problem, a simple MRSLS is developed, and its flowchart is shown in Figure 7.17 Flowchart of MRSLS.. A solution is represented by three vectors (MAV, RAV, WAV), following the similar encoding scheme presented in section 4.5.1. The algorithm starts by creating a random solution. In this case, the method is described in section 4.5.2. The solution then is improved by either a pairwise exchange in the positions of the vectors (MAV, RAV, WAV) or a change in the resources assignment (MAV and WAV). If at least one of the objectives is better than the previous one, then the improved method is repeated until no further improved is achieved. The solution current solution is saved. After that, a new solution is generated and then improve method is executed. The algorithm will stop until a maximum time is achieved. At the end of the algorithm, a non-dominated sort is executed, and the best Pareto front is retrieved.

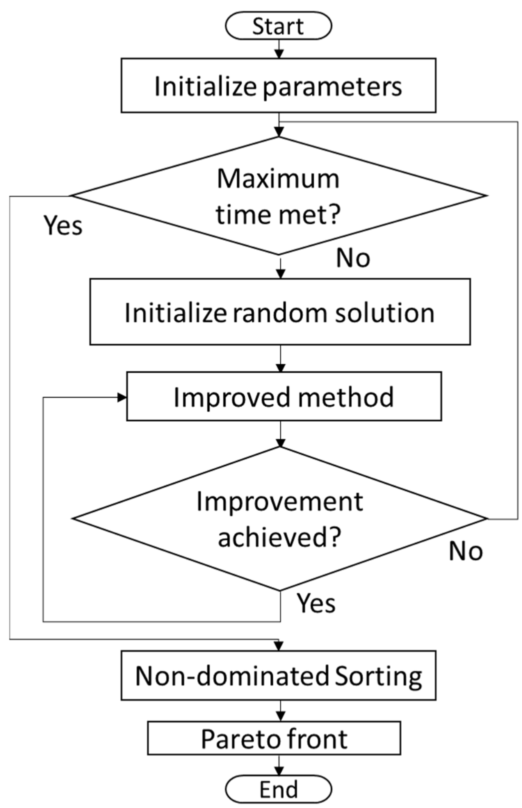



*Figure 7.17 Flowchart of MRSLS.*

### 7.3.2 Comparison of NSGA-II and the MRSLS

The objective of every multi-objective algorithm is to find a set of non-dominated solutions. The algorithms need to be executed several times. All the solutions generated in the different runs are then combined in a set where a non-dominated sort is executed. Finally, the efficient Pareto front solutions are reported.

In this experimentation, the parameters of the NSGA-II are the ones found during the design of experiment section. The NSGA-II is executed four times, and the non-dominated solutions of each execution are saved. In the same manner, the computational time is saved too. The solutions are sorted, and the efficient Pareto front of this combined set is provided. The average execution time is calculated which will be used to set the maximum time for the MRSLS.

A similar approach will be performed with MRSLS and the time will be set as the average time of the NSGA-II execution. Finally, a comparison is made with the union of the solution coming from NSGA-II and MRSLS. The percentage of contribution of the final non-dominated set is reported.

The results presented in Table 7.17 reveals that the NSGA-II outperformed the MRSLS for instance DR1. NSGA-II provided a total of 7 non-dominated solutions, and the Gantt charts of the machines and workers for the Pareto-optimal solutions of the NSGA-II for DR1 are shown in Figure 7.18 to Figure 7.31.

*Table 7.17 Comparison of the NSGA-II and the MRSLS (DR1).*

| DR1 | NSGA-II | | | MRSLS | | |
|---|---|---|---|---|---|---|
| Solution | $c_{max}$ | $\sum_{j=1}^{n} W_j T_j$ | $L_{max}$ | $c_{max}$ | $\sum_{j=1}^{n} W_j T_j$ | $L_{max}$ |
| 1 | 68 | 641 | 45 | 95 | 956 | 63 |
| 2 | 65 | 660 | 45 | 86 | 1057 | 61 |
| 3 | 69 | 597 | 45 | | | |
| 4 | 71 | 562 | 45 | | | |
| 5 | 66 | 656 | 45 | | | |
| 6 | 70 | 575 | 45 | | | |
| 7 | 67 | 644 | 45 | | | |
| Average time | 138.51 s | | | | | |

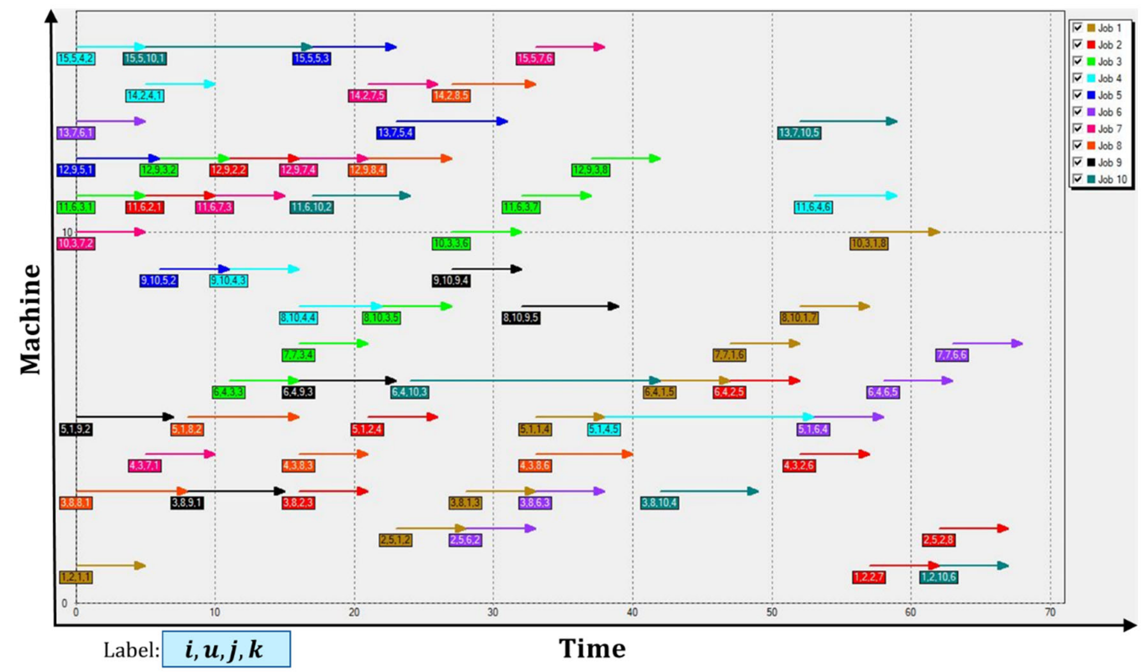



*Figure 7.18 Gantt chart of machines of DR1-solution 1.*

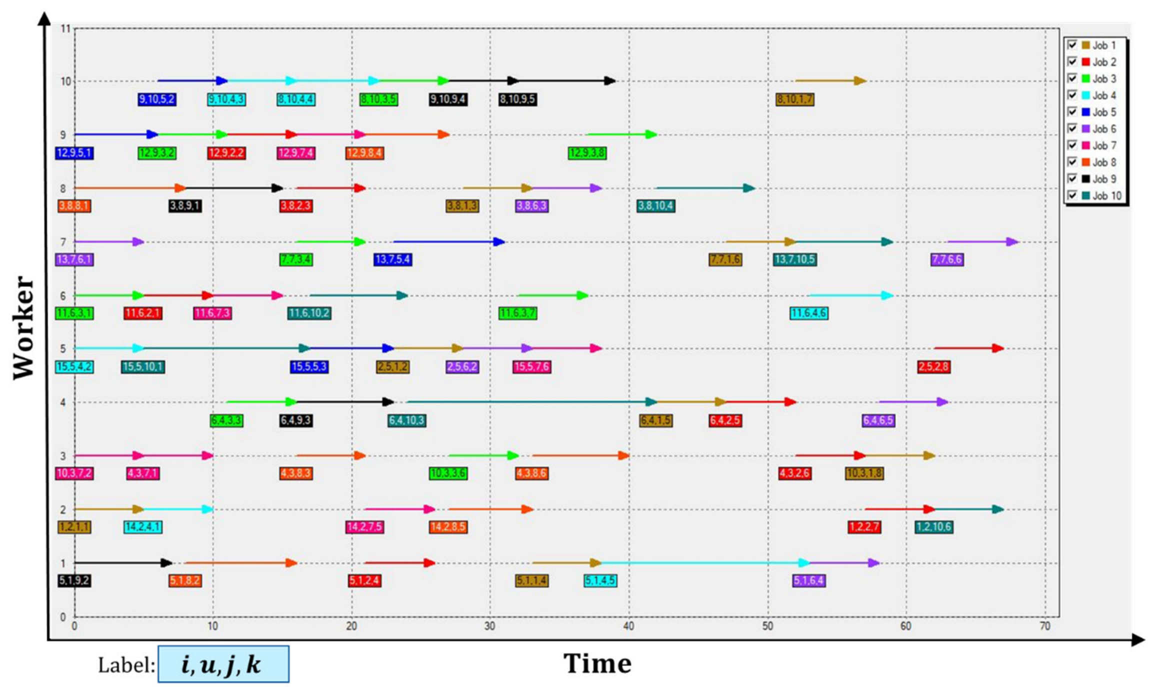



*Figure 7.19 Gantt chart of workers of DR1-solution 1.*

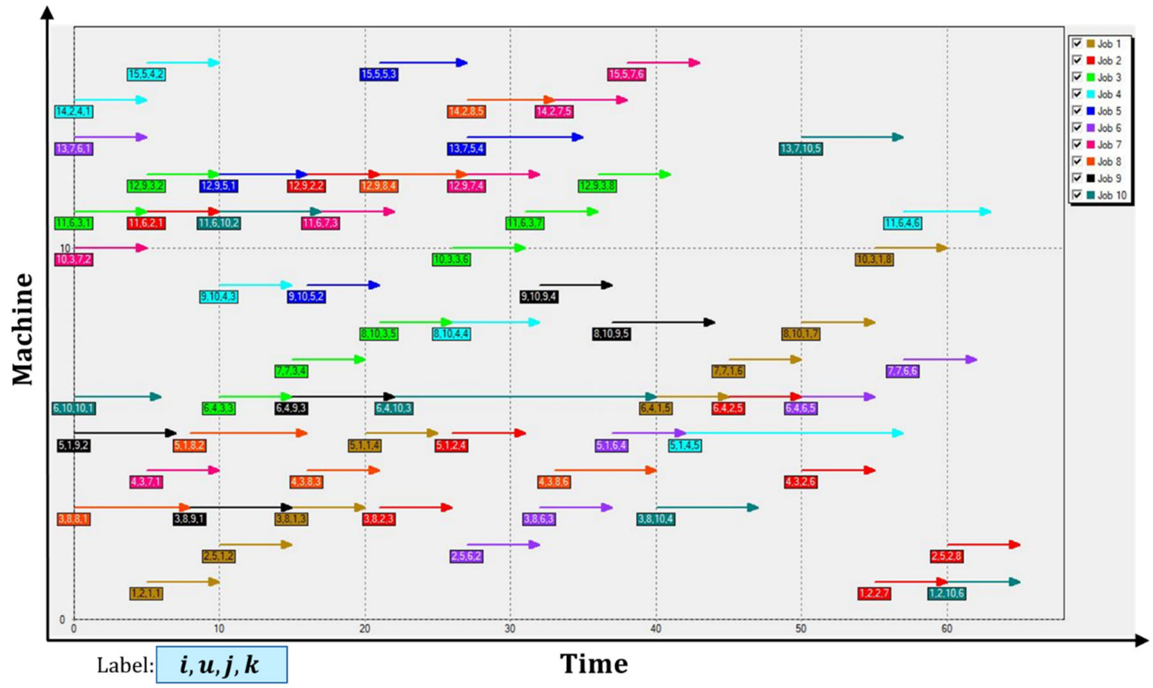


*Figure 7.20 Gantt chart of machines of DR1-solution 2.*

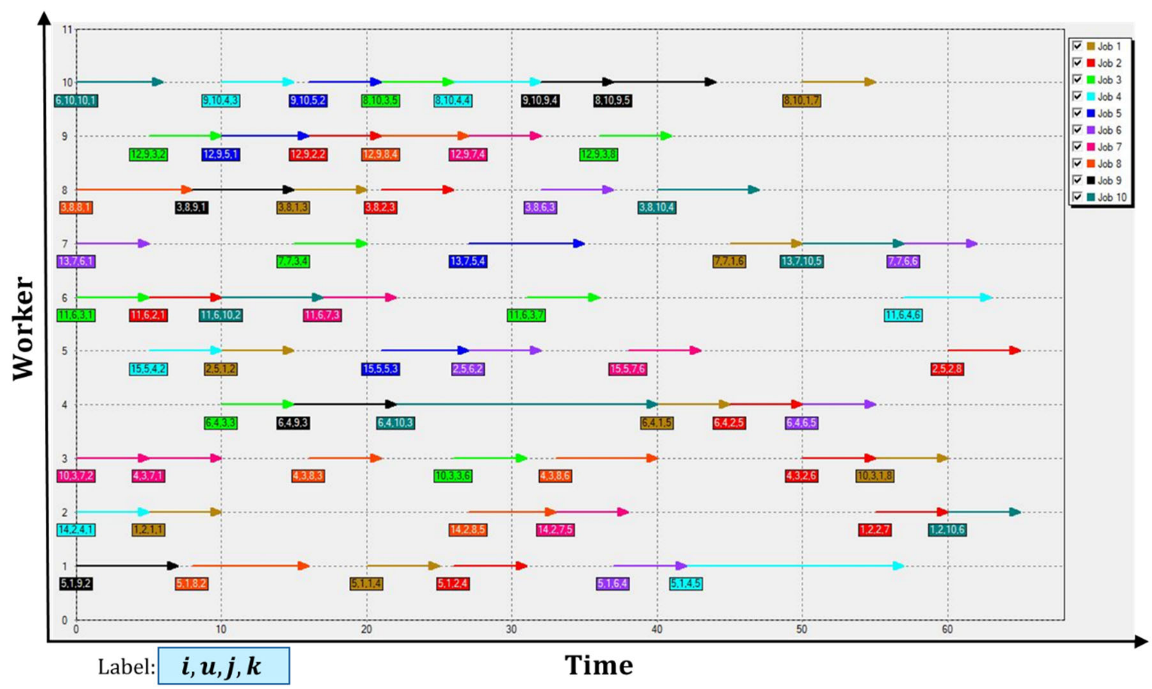


*Figure 7.21 Gantt chart of workers of DR1-solution 2.*

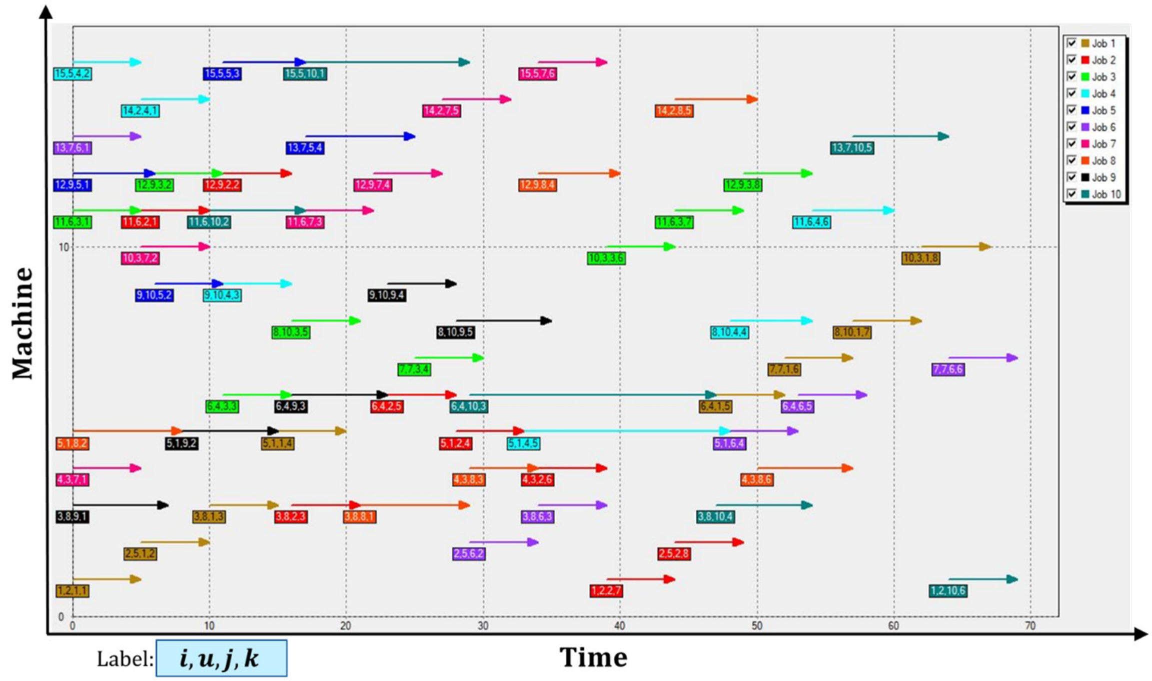


*Figure 7.22 Gantt chart of machines of DR1-solution 3.*

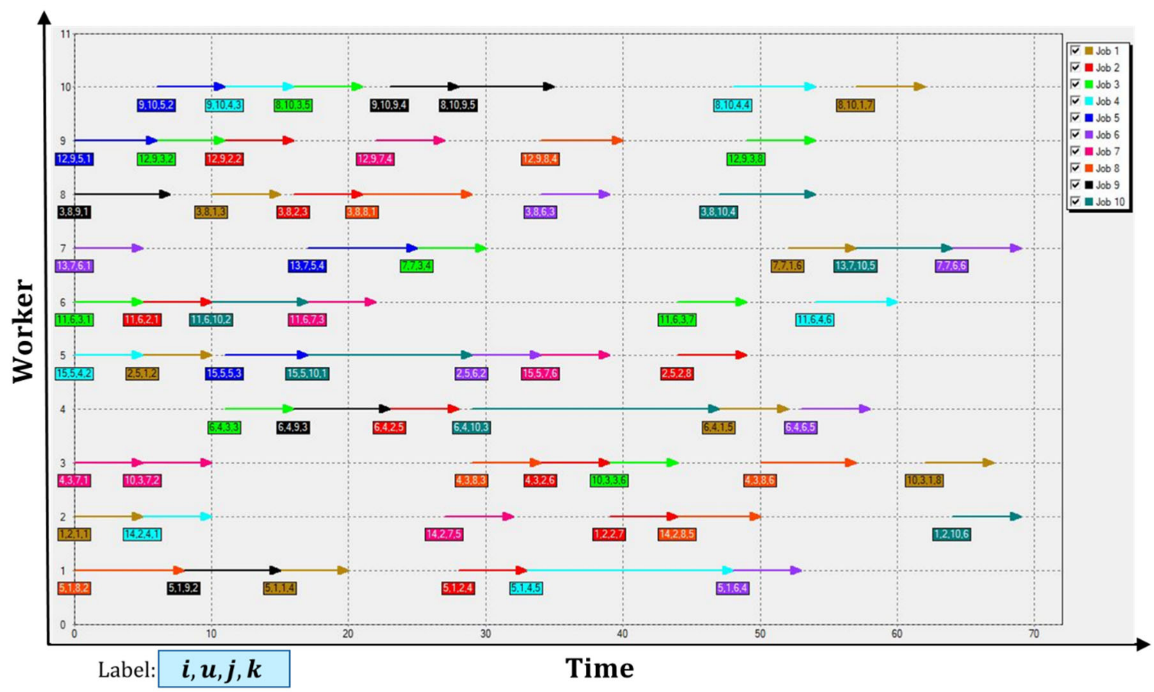


*Figure 7.23 Gantt chart of workers of DR1-solution 3.*

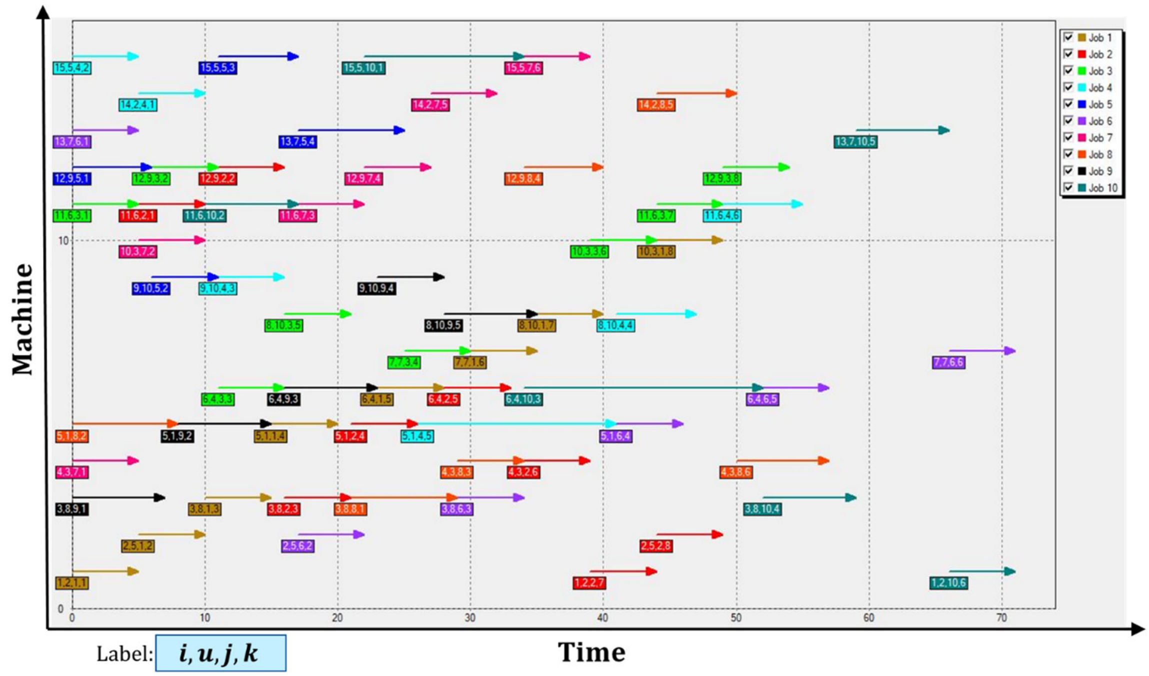


*Figure 7.24 Gantt chart of machines of DR1-solution 4.*

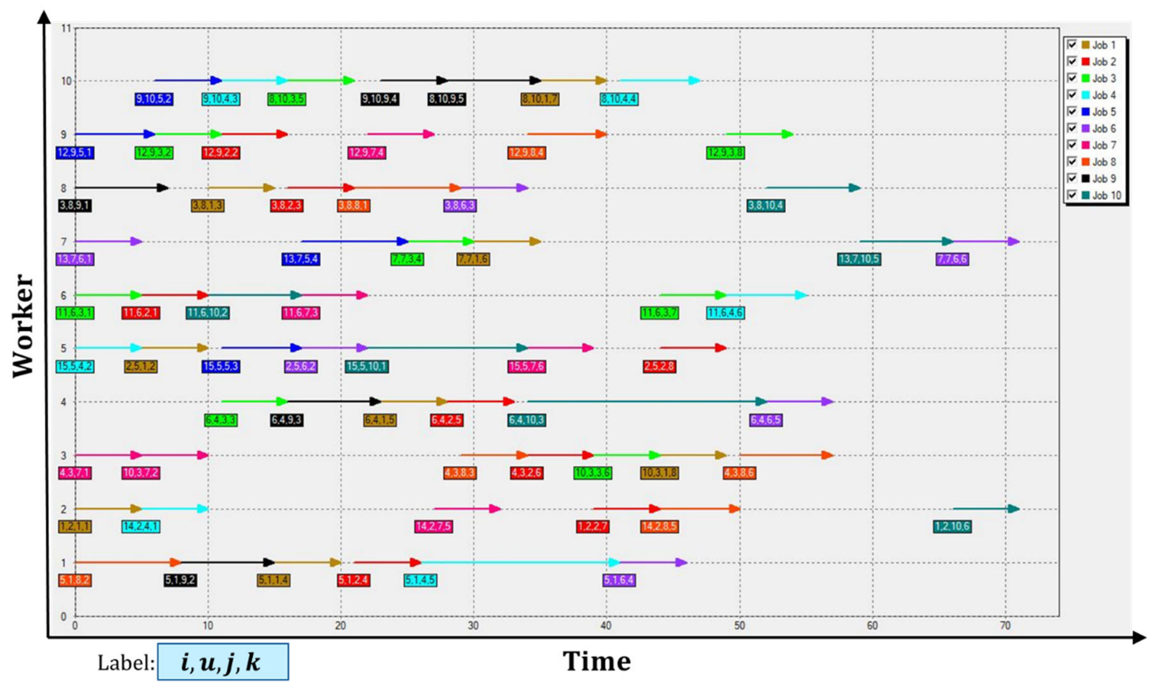


*Figure 7.25 Gantt chart of workers of DR1-solution 4.*

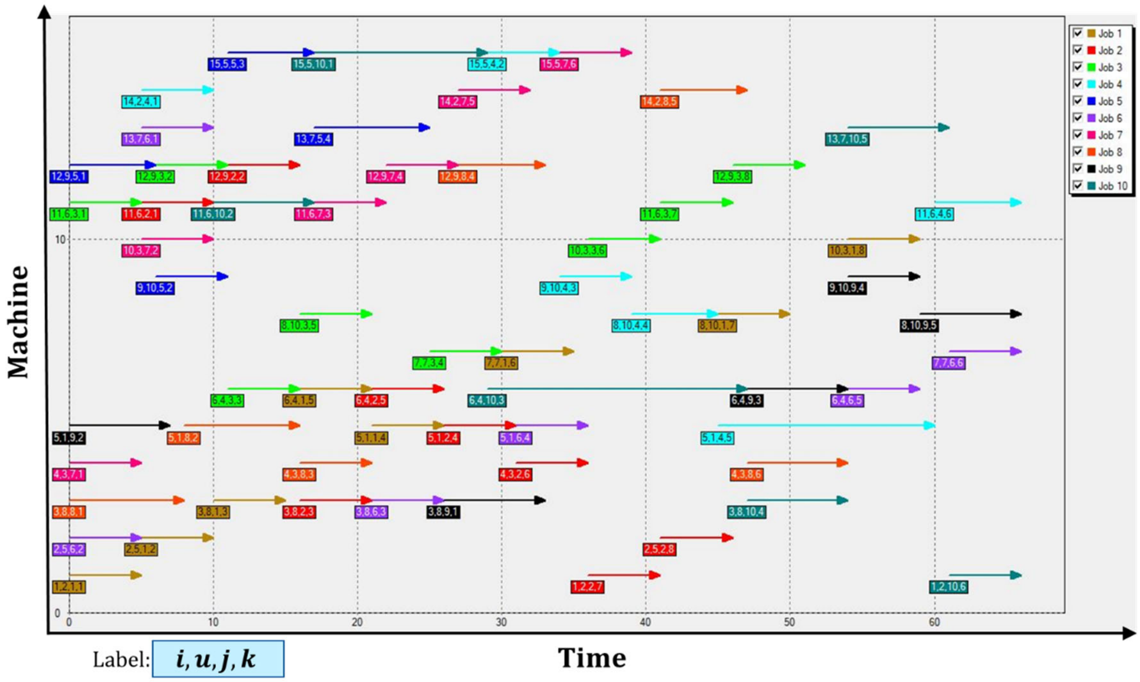



*Figure 7.26 Gantt chart of machines of DR1-solution 5.*

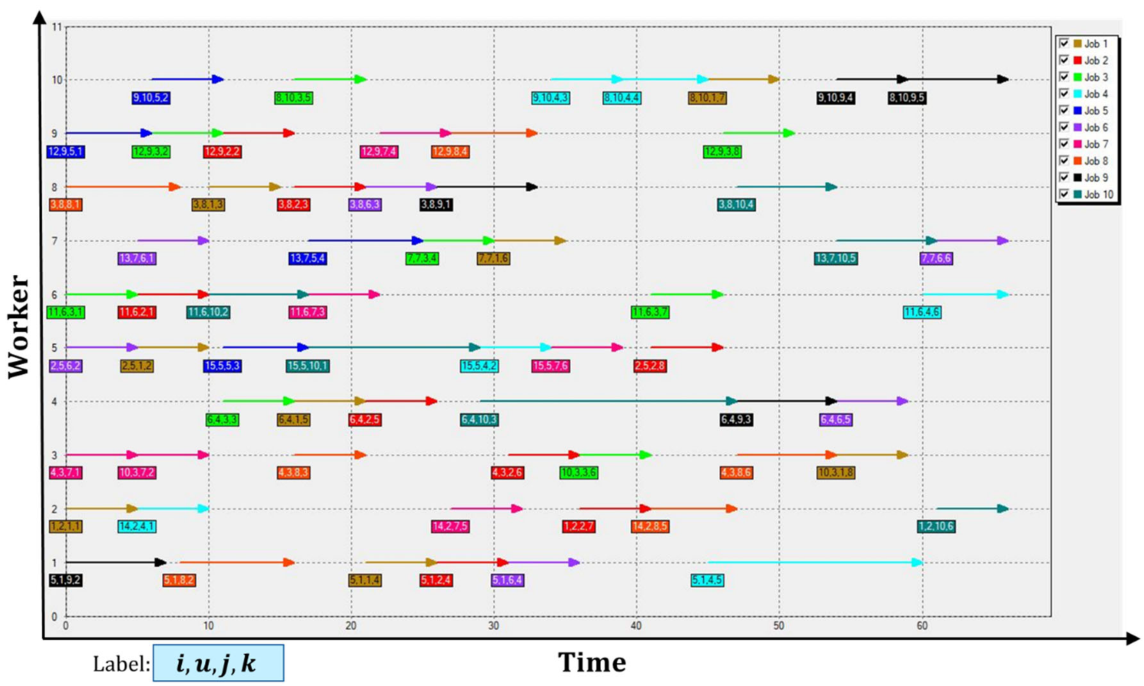



*Figure 7.27 Gantt chart of workers of DR1-solution 5.*

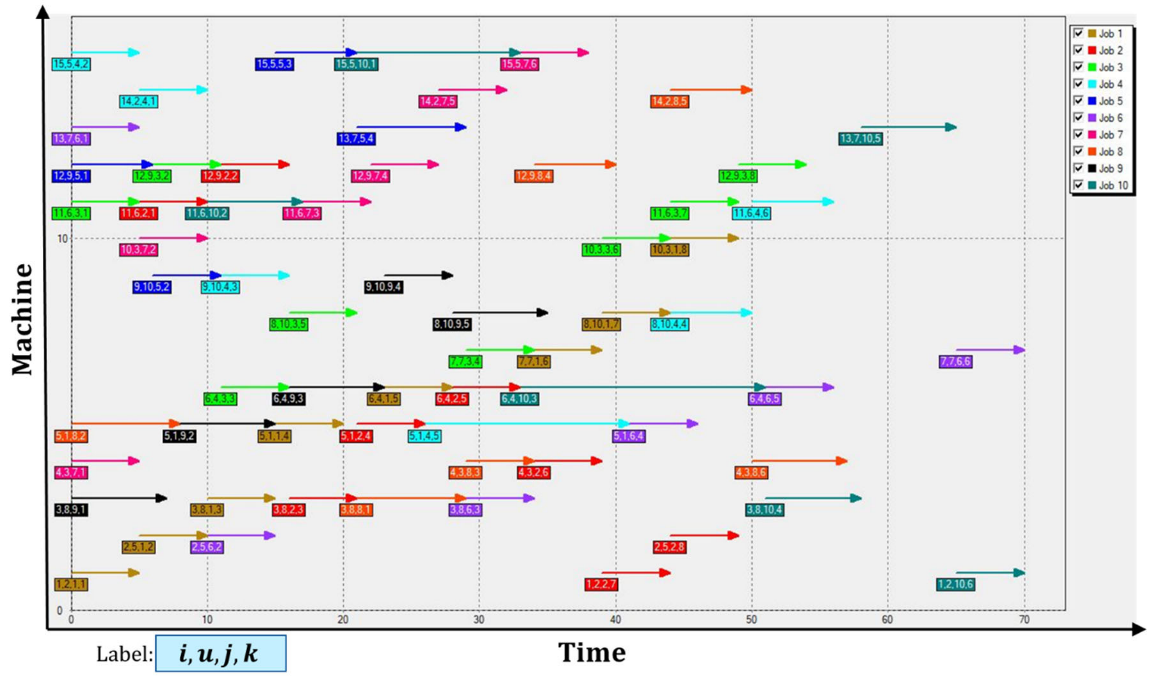



*Figure 7.28 Gantt chart of machines of DR1-solution 6.*

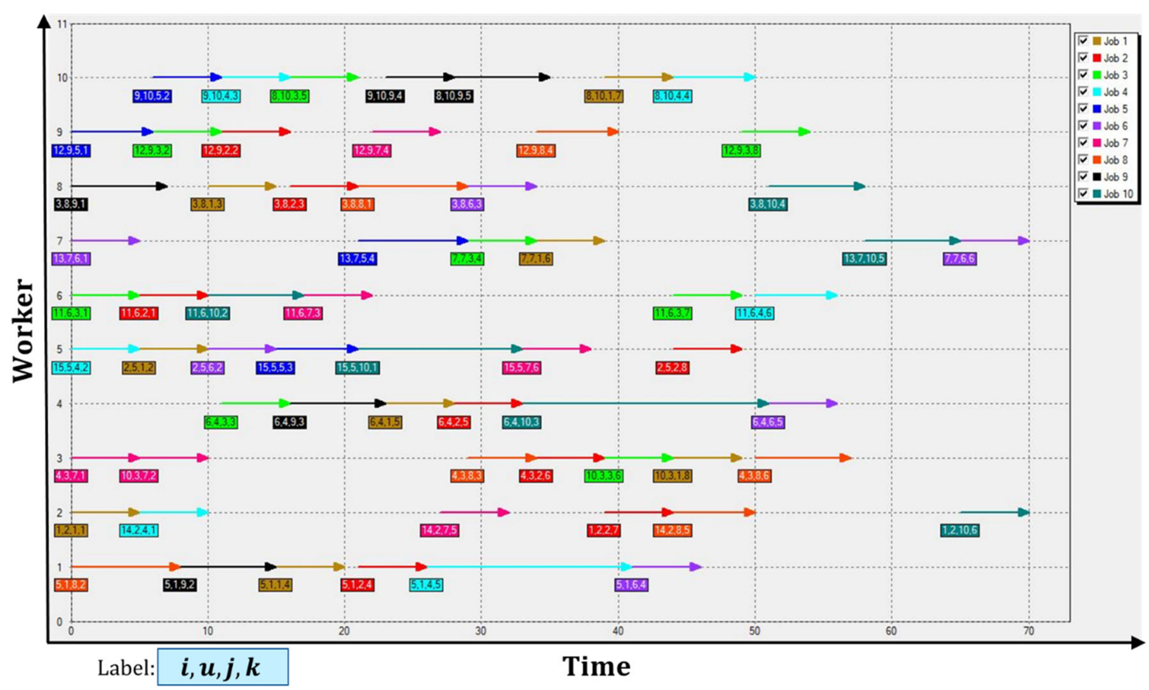



*Figure 7.29 Gantt chart of workers of DR1-solution 6.*

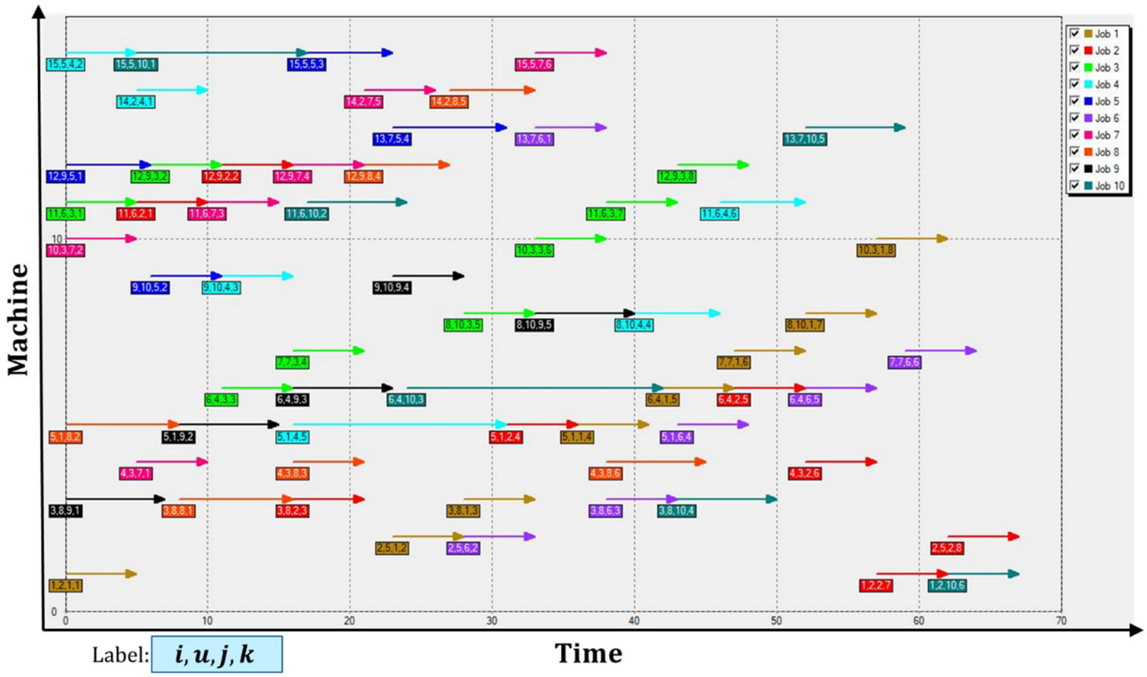



*Figure 7.30 Gantt chart of machines of DR1-solution 7.*

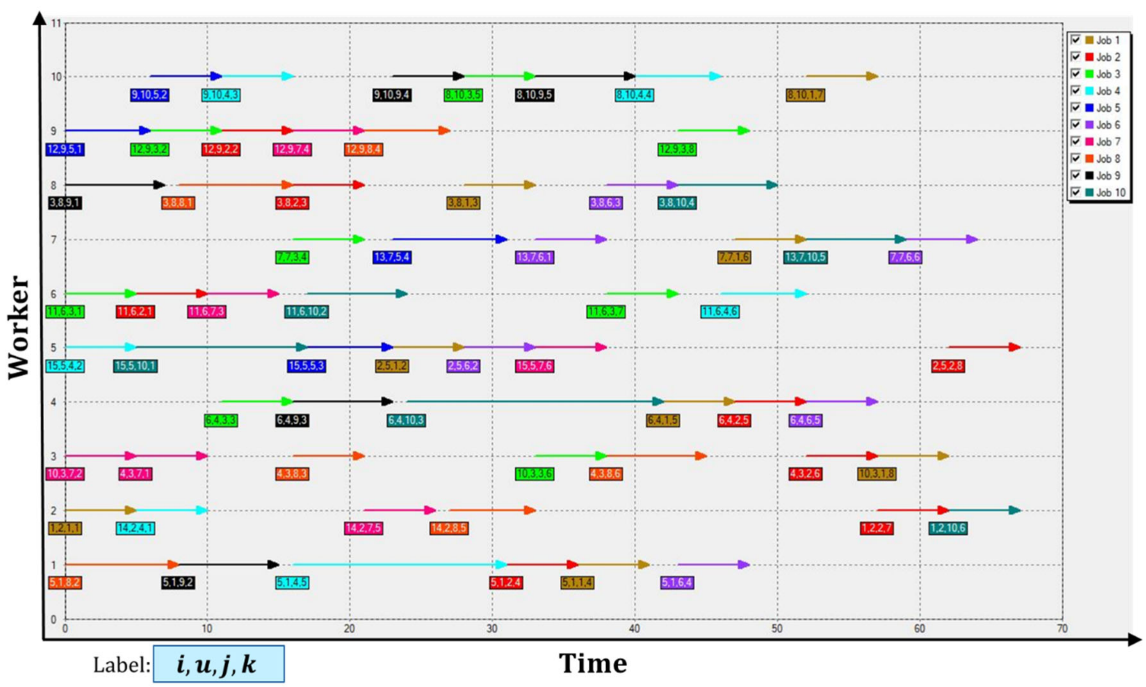



*Figure 7.31 Gantt chart of workers of DR1-solution 7.*

The results presented in Table 7.18 reveals that the NSGA-II outperformed the MRSLS for the instance DR2. NSGA-II provided two non-dominated solutions. These solutions dominate the ones provided by MRSLS. Gantt charts of the machines and workers for the Pareto-optimal solutions of the NSGA-II for DR1 are presented in Figure 7.32 to Figure 7.35.

*Table 7.18 Comparison of the NSGA-II and the MRSLS (DR2).*

| DR2 | NSGA-II | | | MRSLS | | |
|---|---|---|---|---|---|---|
| Solution | $c_{max}$ | $\sum_{j=1}^{n} W_j T_j$ | $L_{max}$ | $c_{max}$ | $\sum_{j=1}^{n} W_j T_j$ | $L_{max}$ |
| 1 | 22 | 10 | 21 | 36 | 198 | 32 |
| 2 | 21 | 18 | 20 | 39 | 234 | 30 |
| Average time | | 22.43 s | | | | |

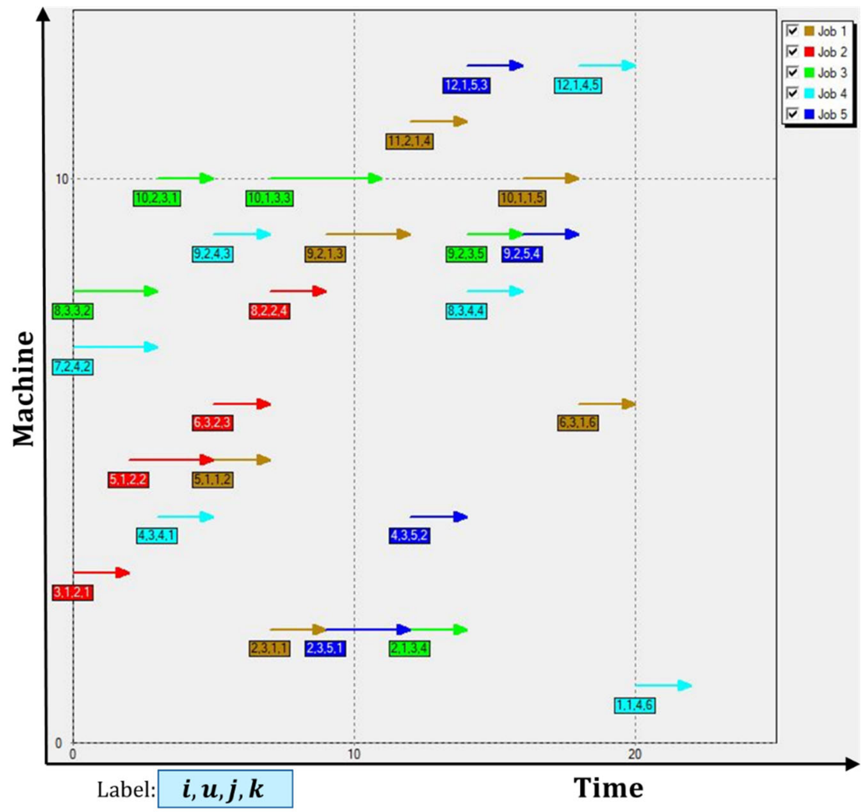



*Figure 7.32 Gantt chart of machines of DR2-solution 1.*

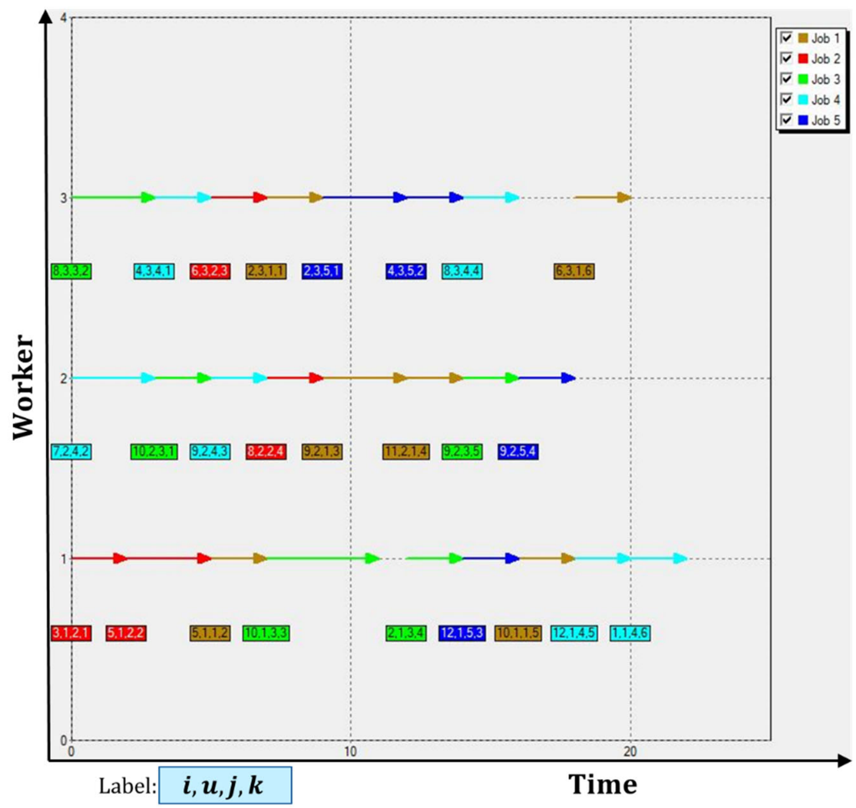



*Figure 7.33 Gantt chart of workers of DR2-solution 1.*

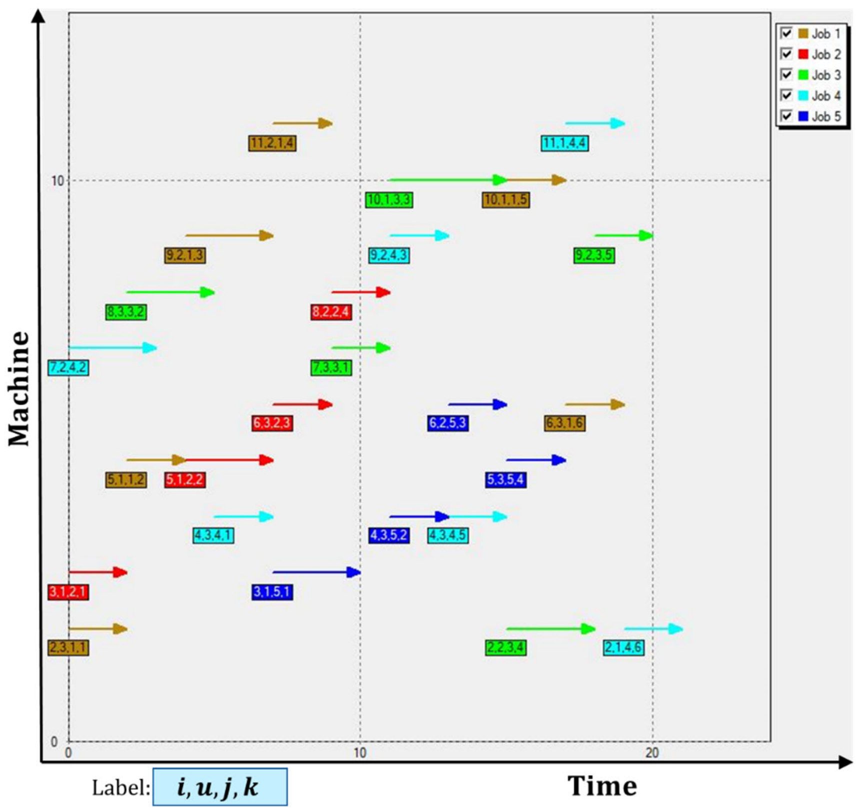



*Figure 7.34 Gantt chart of machines of DR2-solution 2.*

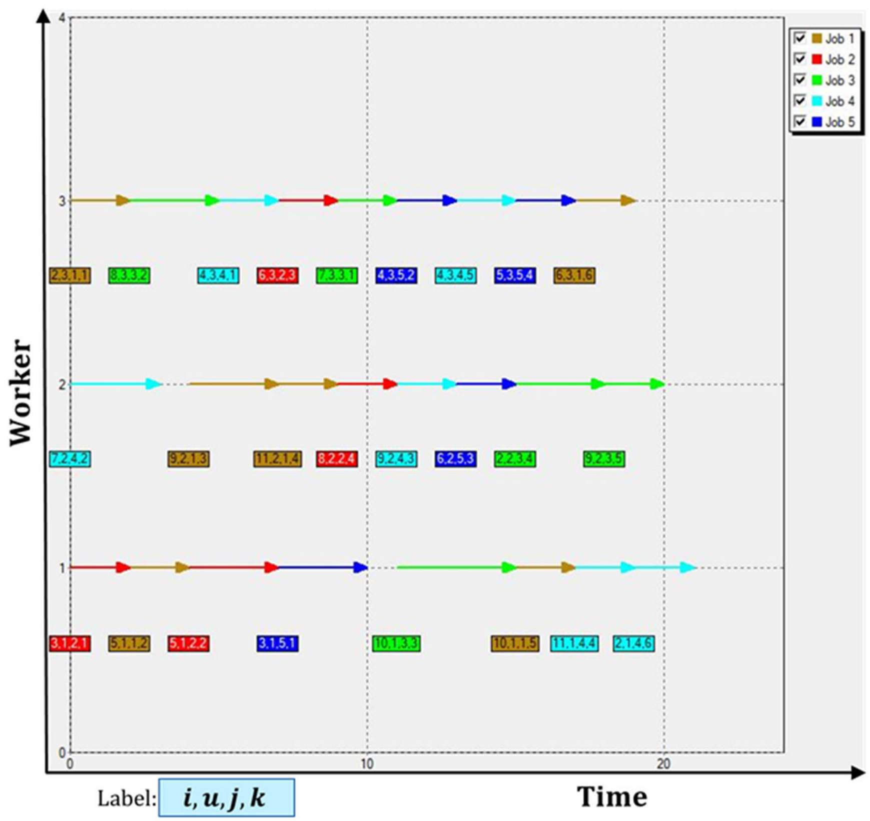



*Figure 7.35 Gantt chart of workers of DR2-solution 2.*

The results presented in Table 7.19 shows that the NSGA-II provided better results than he MRSLS. It provided a total of 9 non-dominated solutions. Gantt charts of the machines and workers for the Pareto-optimal solutions of the NSGA-II for DR3 are shown in Figure 7.36 to Figure 7.53.

*Table 7.19 Comparison of the NSGA-II and the MRSLS (DR3).*

| DR3 | NSGA-II | | | MRSLS | | |
|---|---|---|---|---|---|---|
| Solution | $c_{max}$ | $\sum_{j=1}^{n} W_j T_j$ | $L_{max}$ | $c_{max}$ | $\sum_{j=1}^{n} W_j T_j$ | $L_{max}$ |
| 1 | 56 | 485 | 45 | 105 | 1180 | 70 |
| 2 | 51 | 542 | 45 | 100 | 1555 | 71 |
| 3 | 56 | 524 | 43 | 87 | 1350 | 72 |
| 4 | 52 | 631 | 43 | | | |
| 5 | 54 | 537 | 44 | | | |
| 6 | 52 | 465 | 46 | | | |
| 7 | 53 | 525 | 45 | | | |
| 8 | 50 | 491 | 46 | | | |
| 9 | 58 | 467 | 44 | | | |
| Average time | 44.02 s | | | | | |

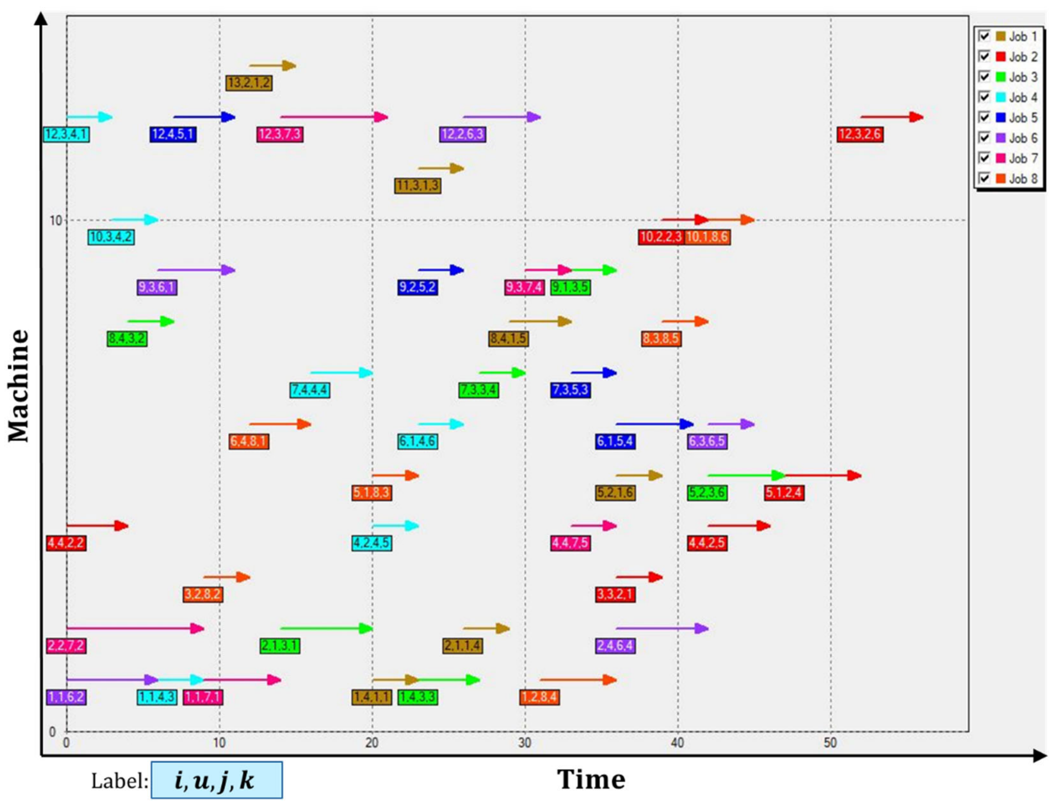



*Figure 7.36 Gantt chart of machines of DR3-solution 1.*

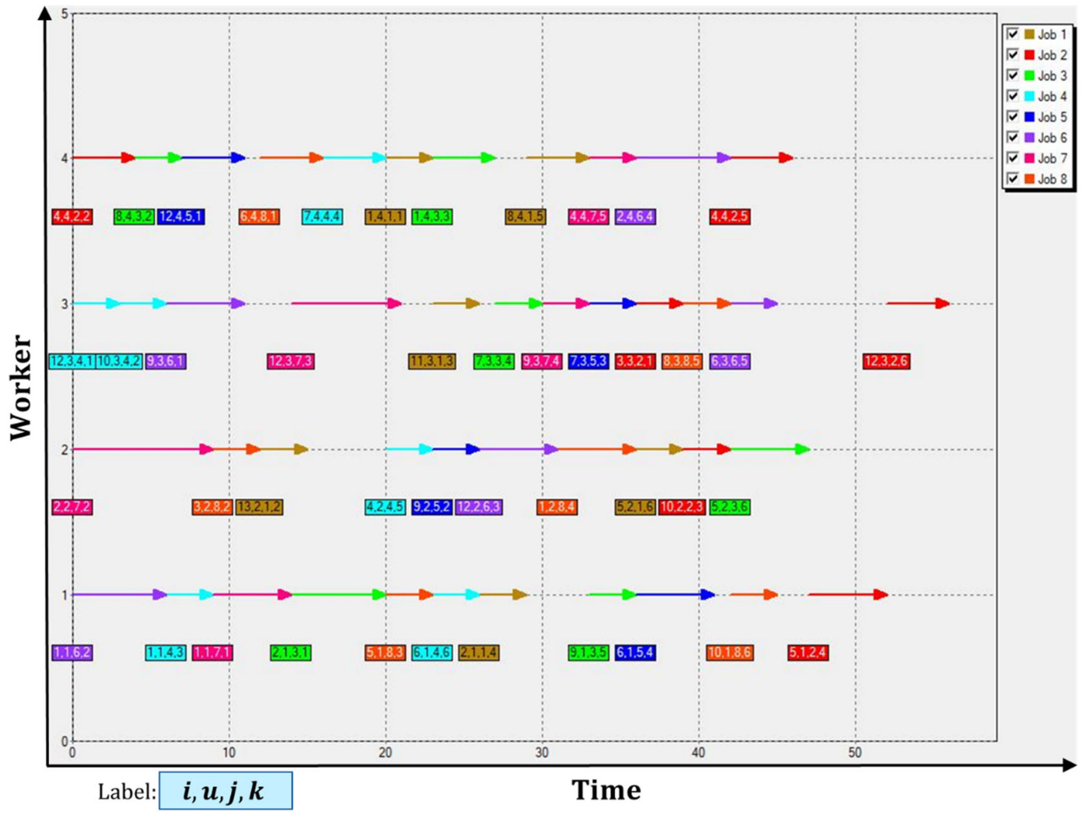



*Figure 7.37 Gantt chart of workers of DR3-solution 1.*

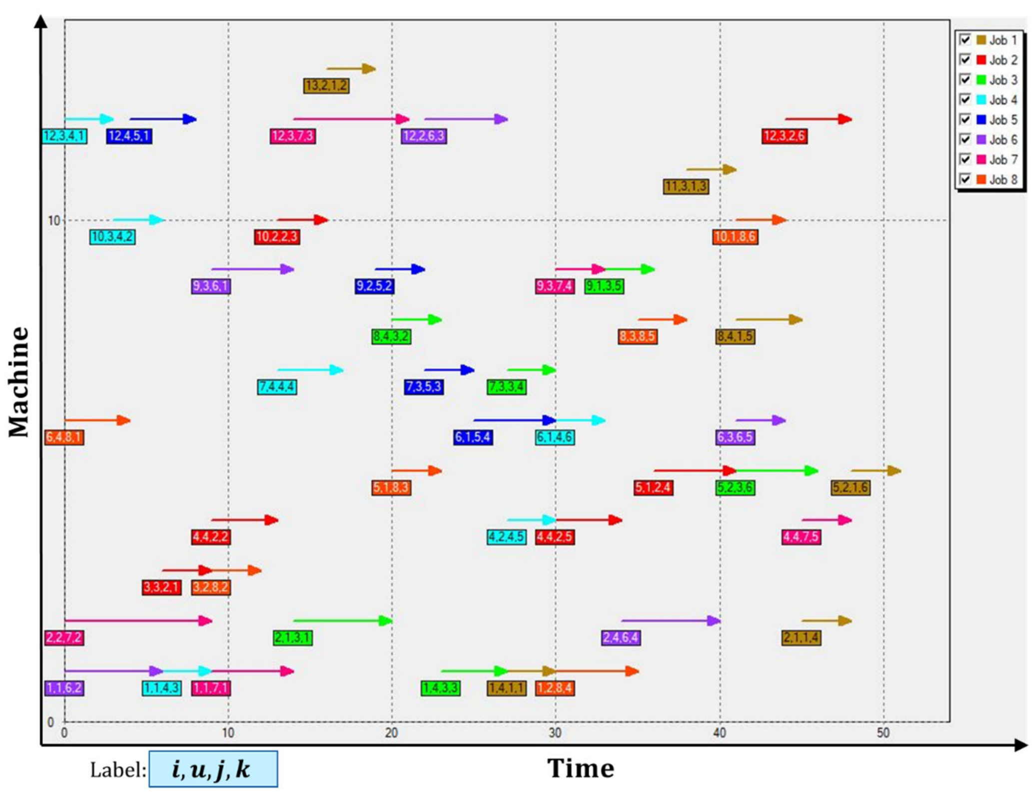



*Figure 7.38 Gantt chart of machines of DR3-solution 2.*

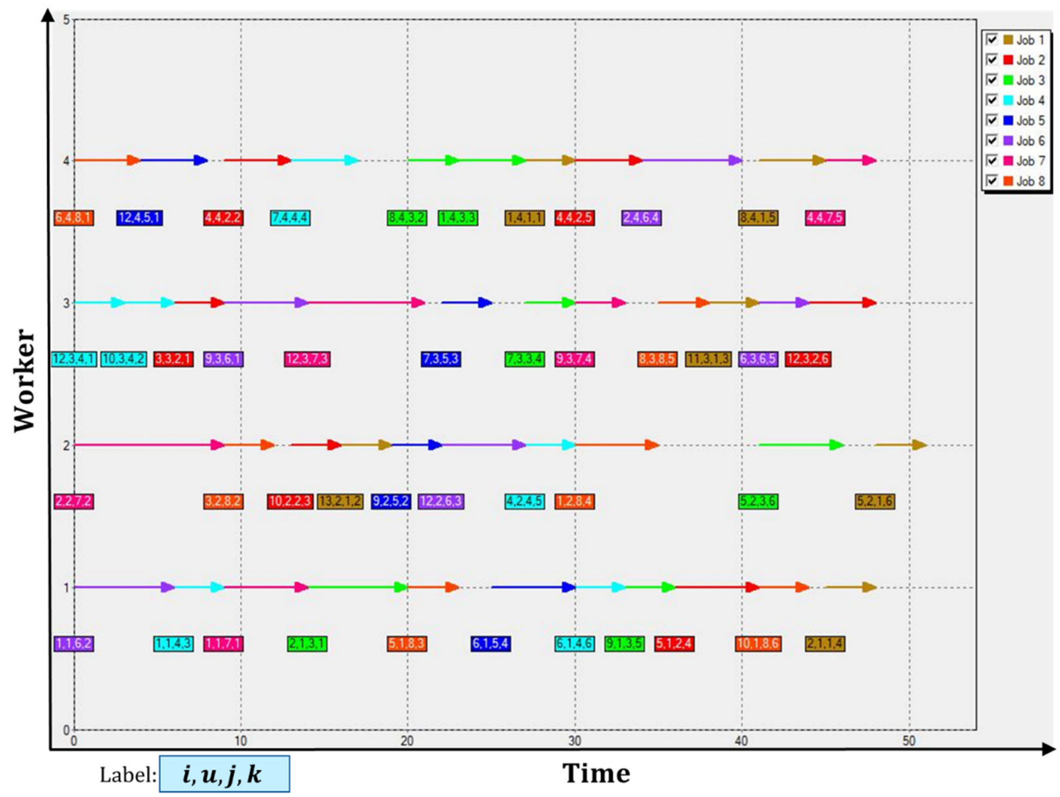



*Figure 7.39 Gantt chart of workers of DR3-solution 2.*

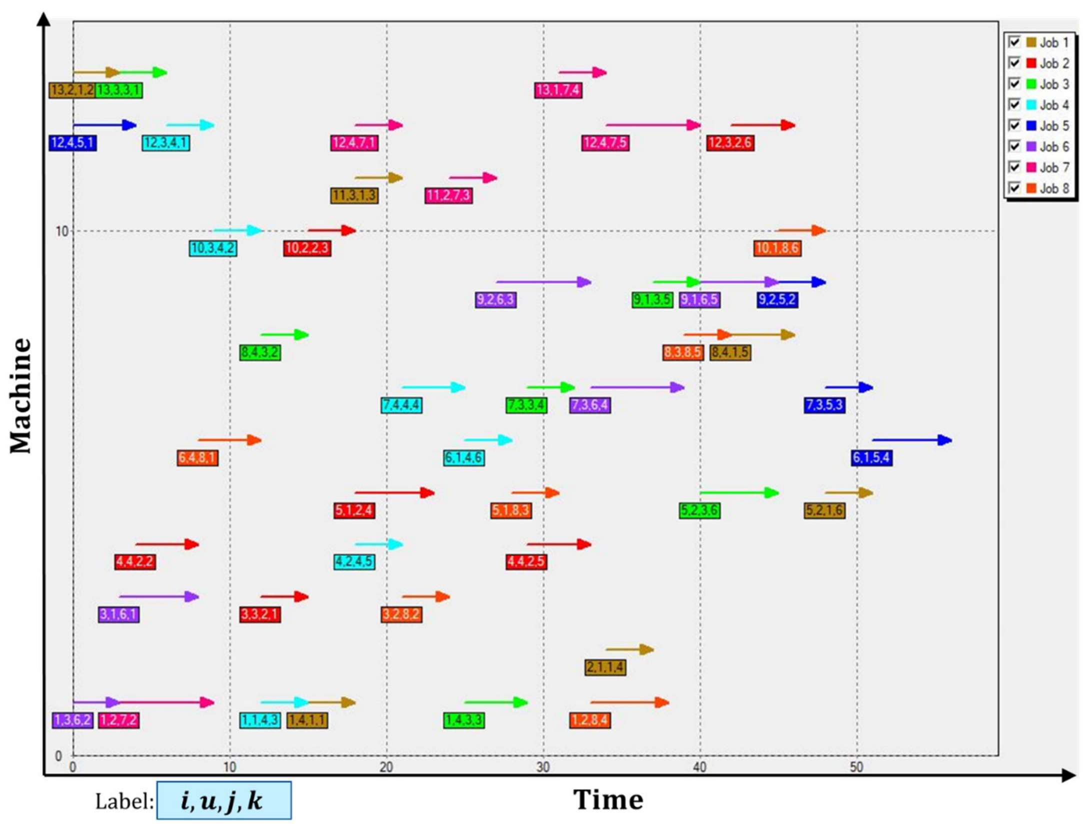



*Figure 7.40 Gantt chart of machines of DR3-solution 3.*

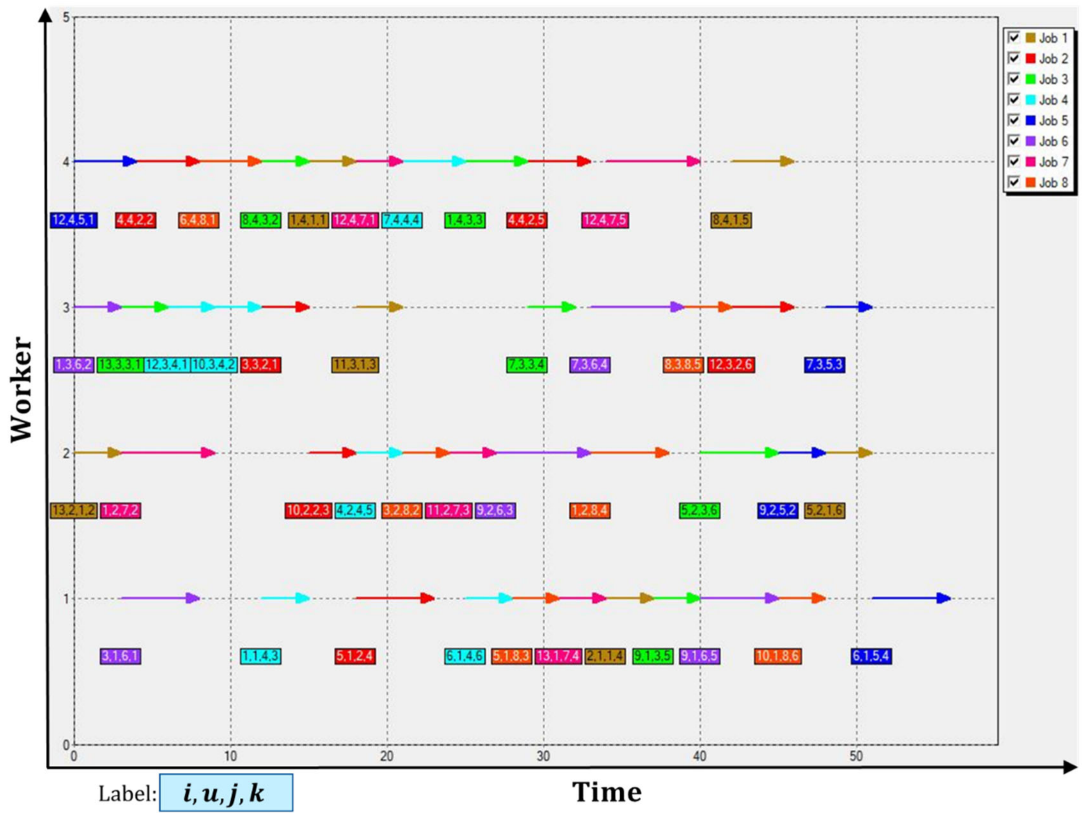


*Figure 7.41 Gantt chart of workers of DR3-solution 3.*

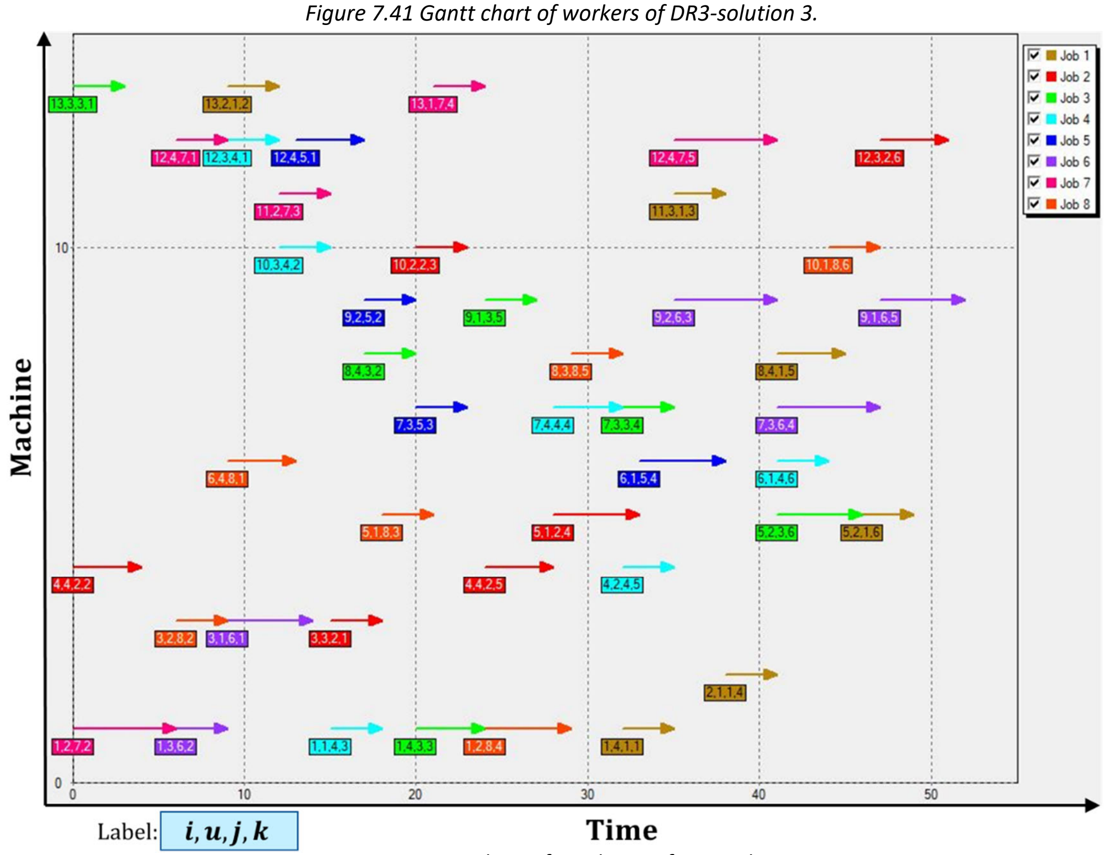


*Figure 7.42 Gantt chart of machines of DR3-solution 4.*

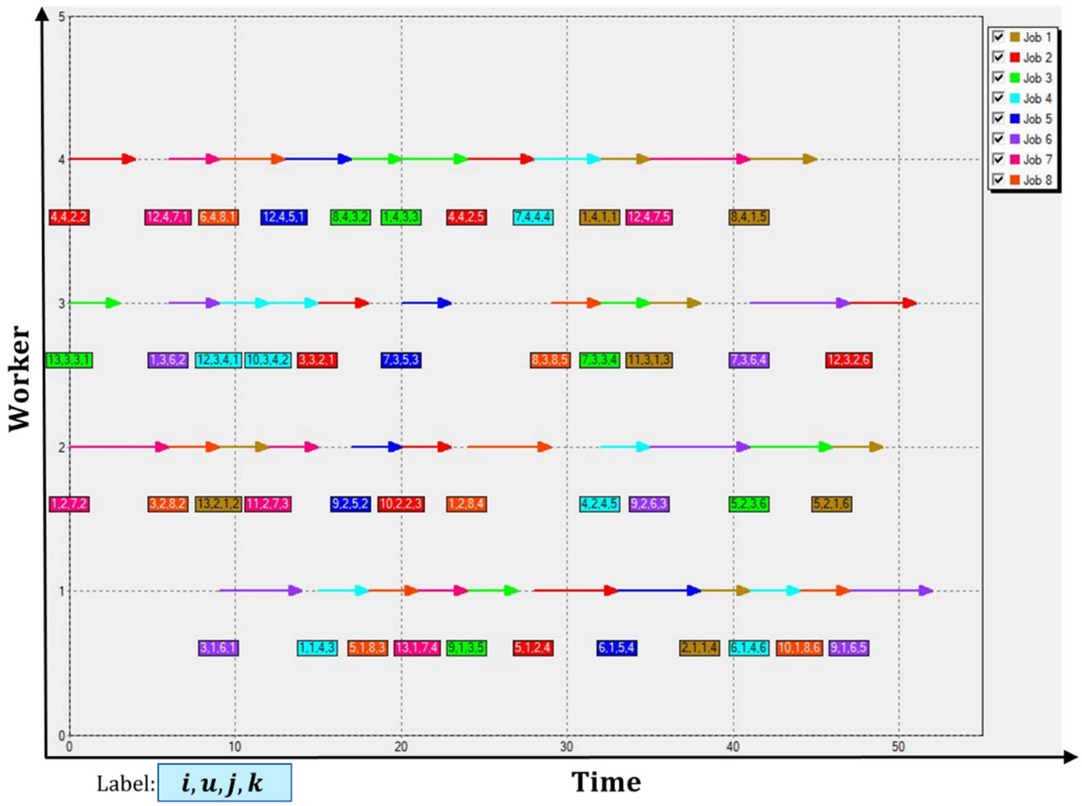


*Figure 7.43 Gantt chart of workers of DR3-solution 4.*

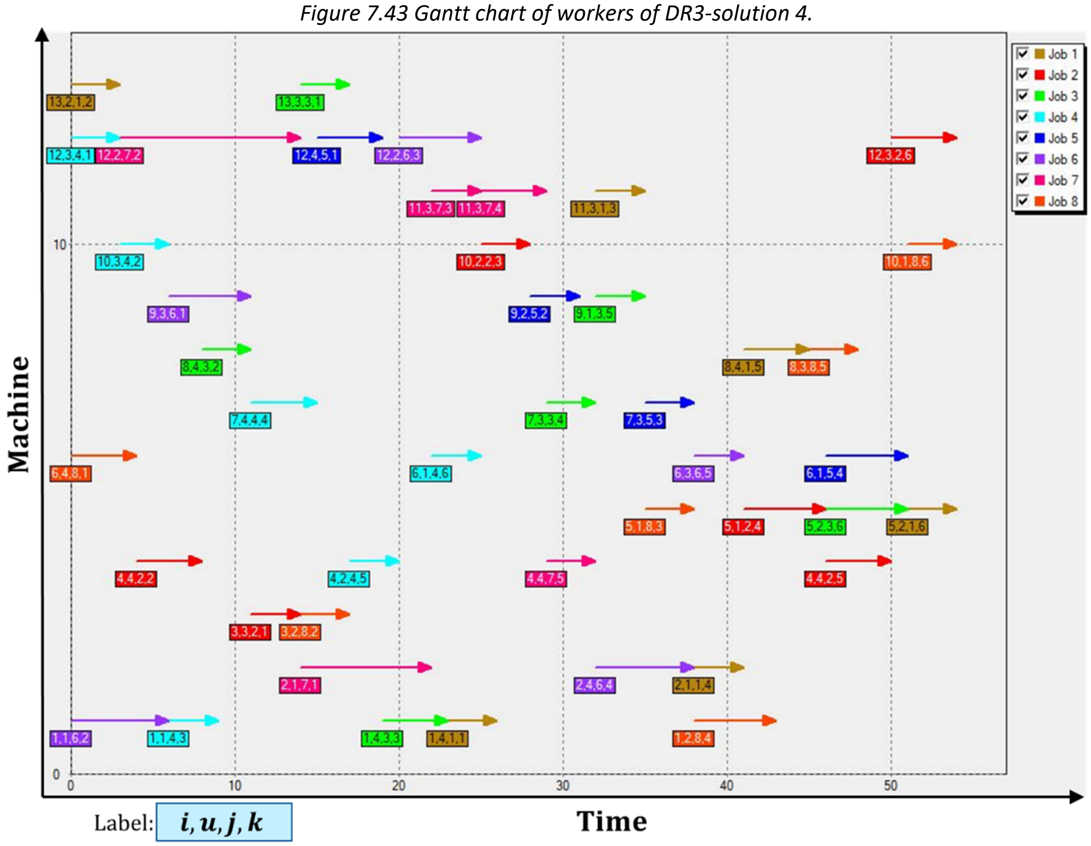


*Figure 7.44 Gantt chart of machines of DR3-solution 5.*

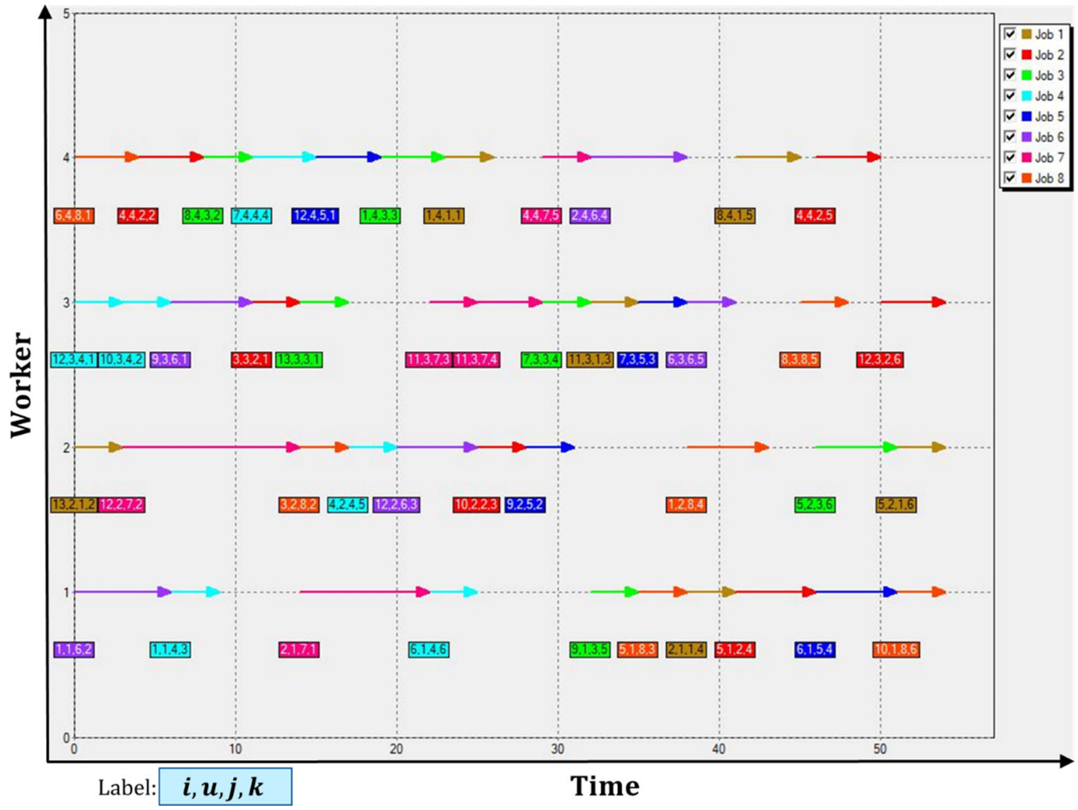



*Figure 7.45 Gantt chart of workers of DR3-solution 5.*

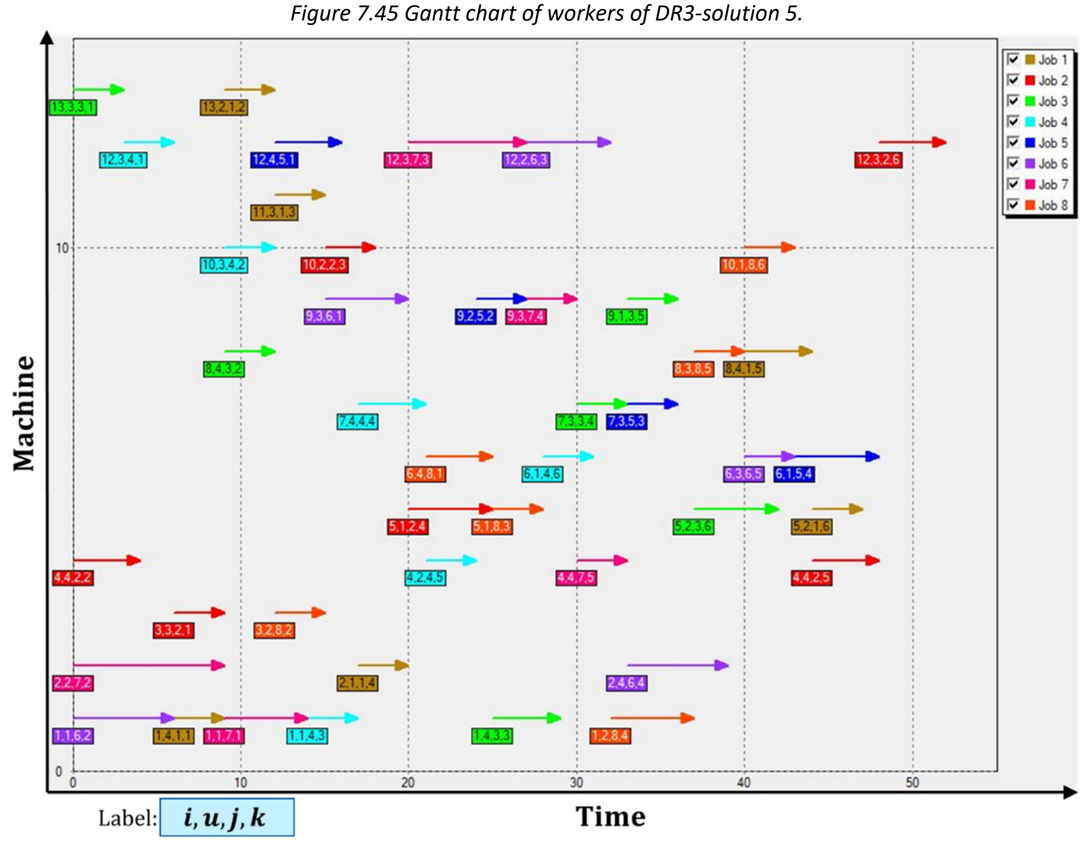



*Figure 7.46 Gantt chart of machines of DR3-solution 6.*

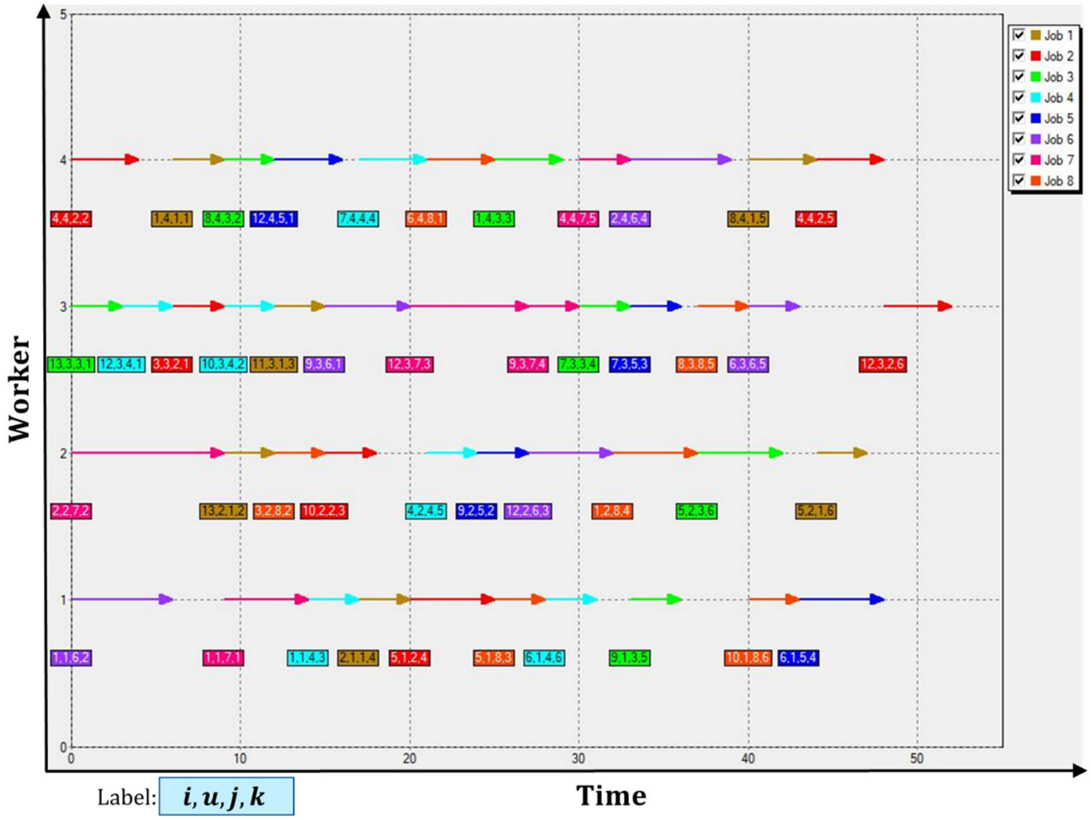



*Figure 7.47 Gantt chart of workers of DR3-solution 6.*

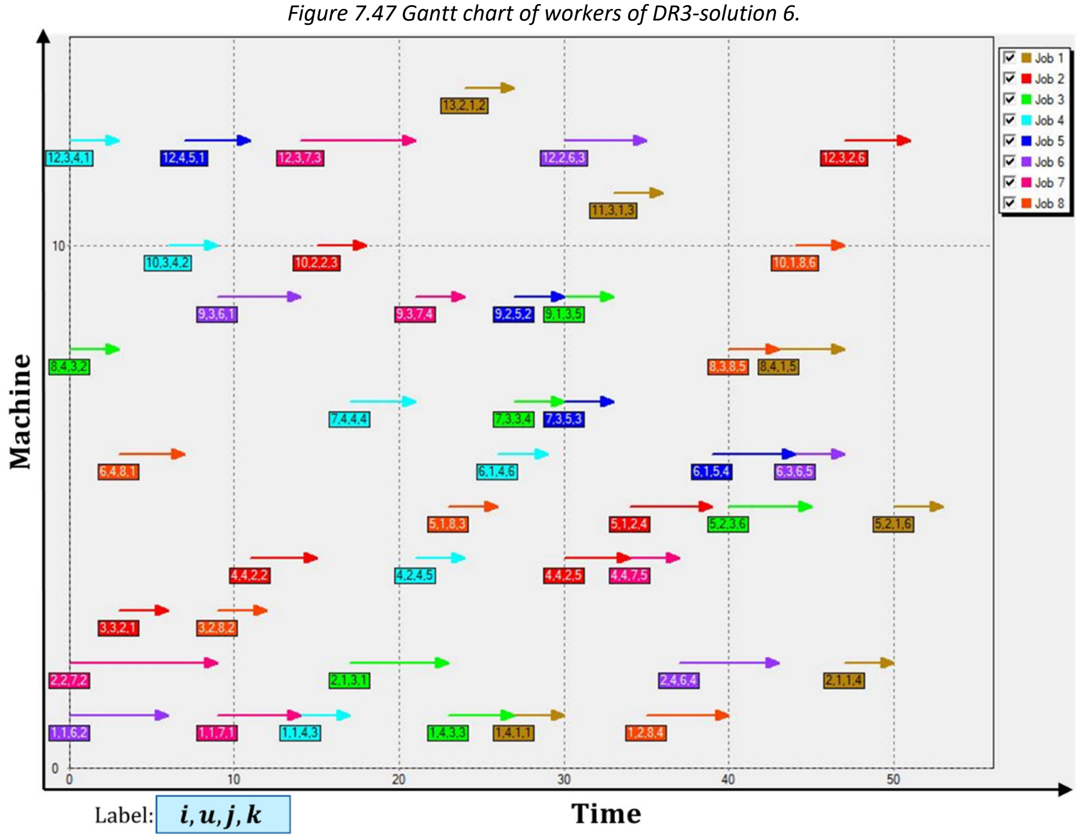



*Figure 7.48 Gantt chart of machines of DR3-solution 7.*

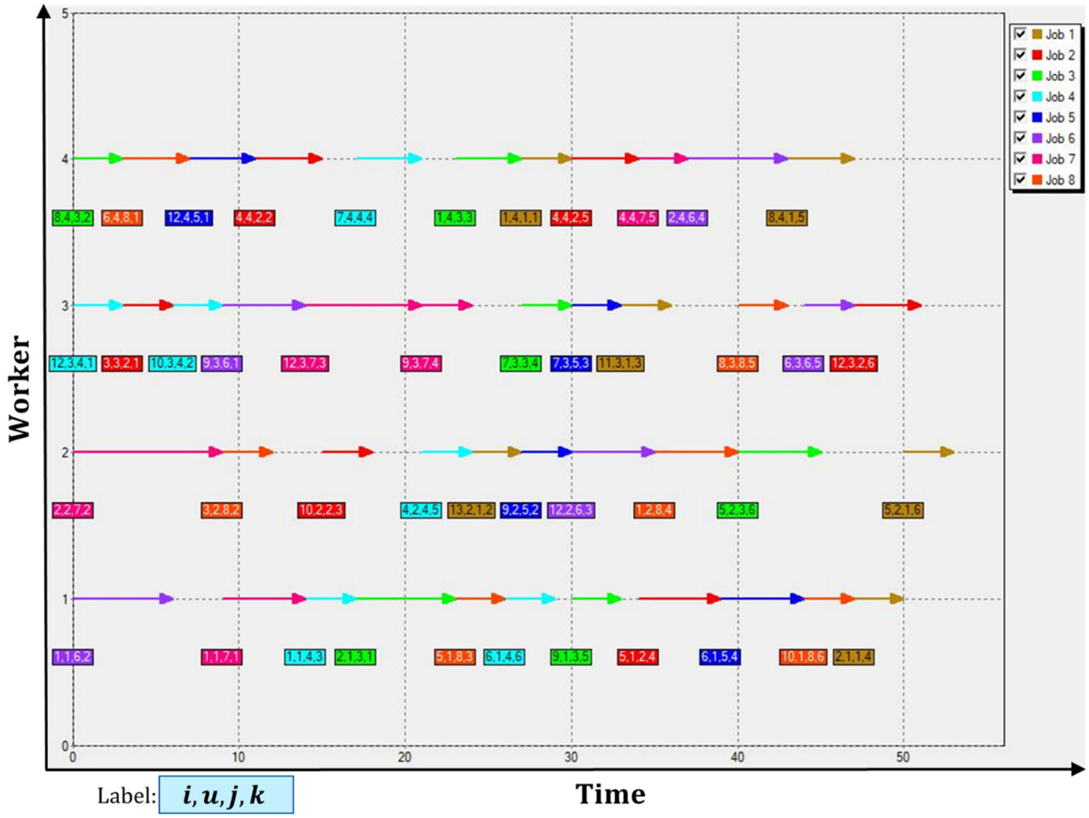



*Figure 7.49 Gantt chart of workers of DR3-solution 7.*

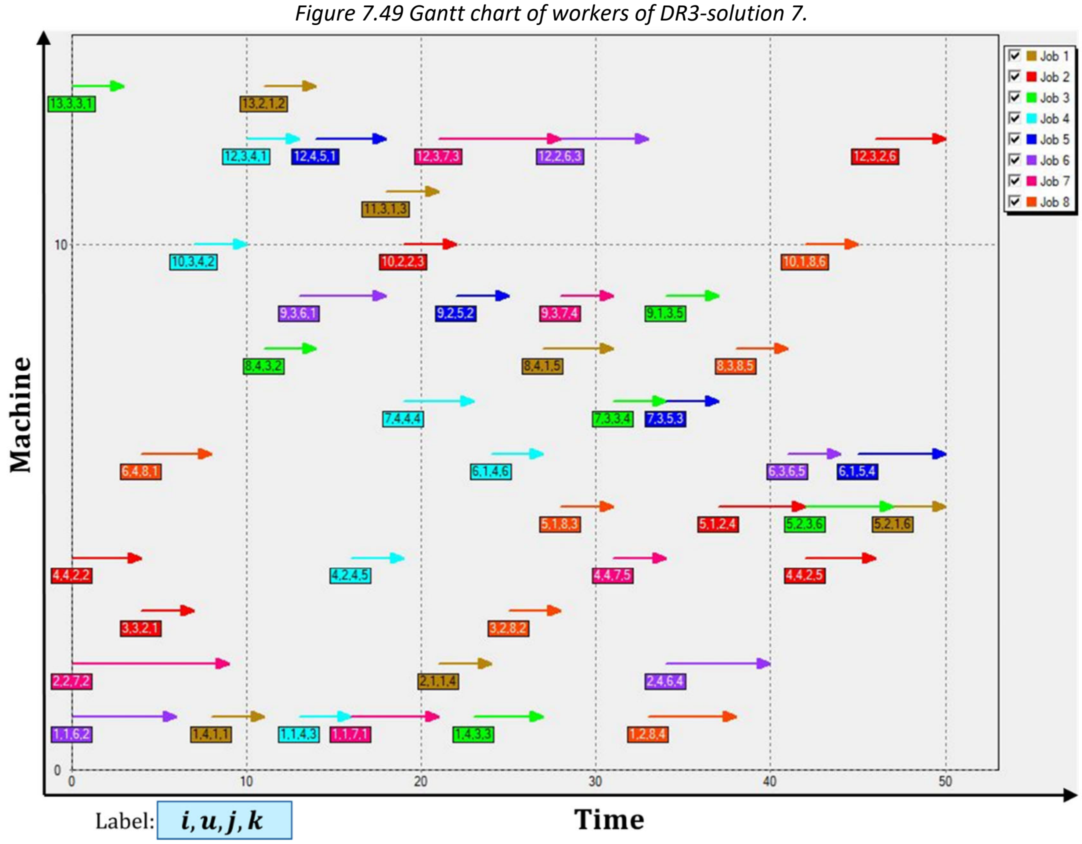



*Figure 7.50 Gantt chart of machines of DR3-solution 8.*

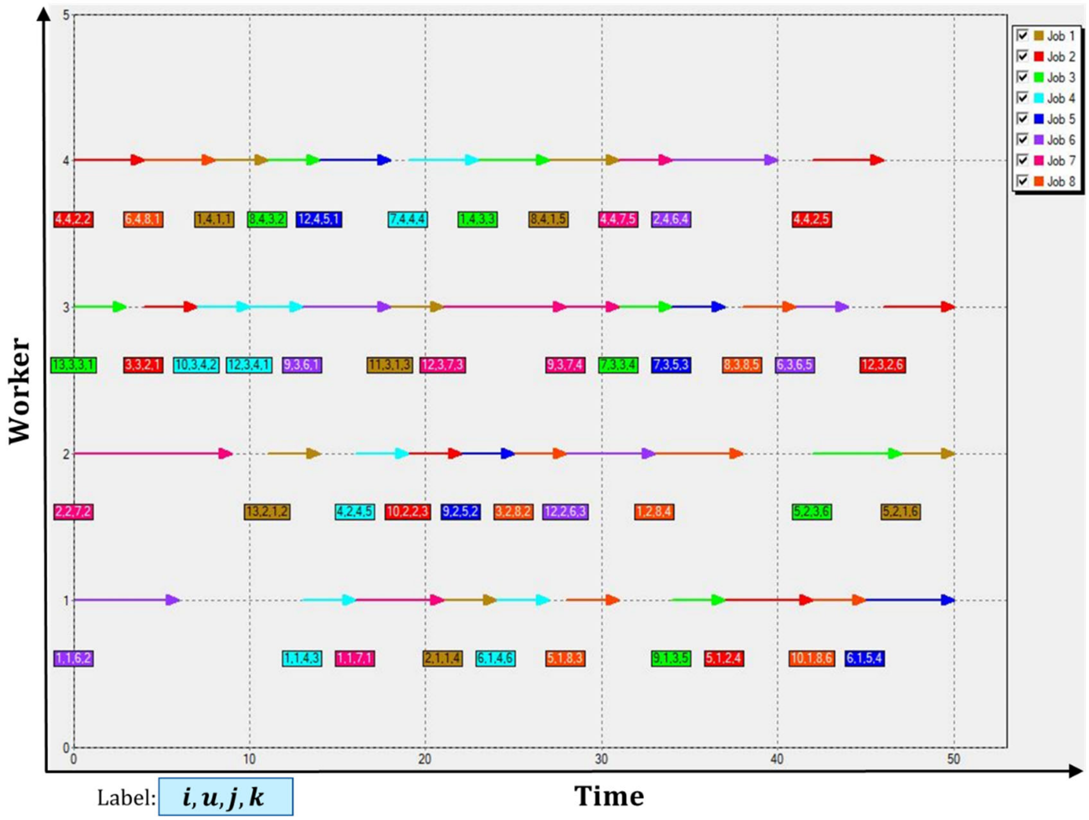


*Figure 7.51 Gantt chart of workers of DR3-solution 8.*

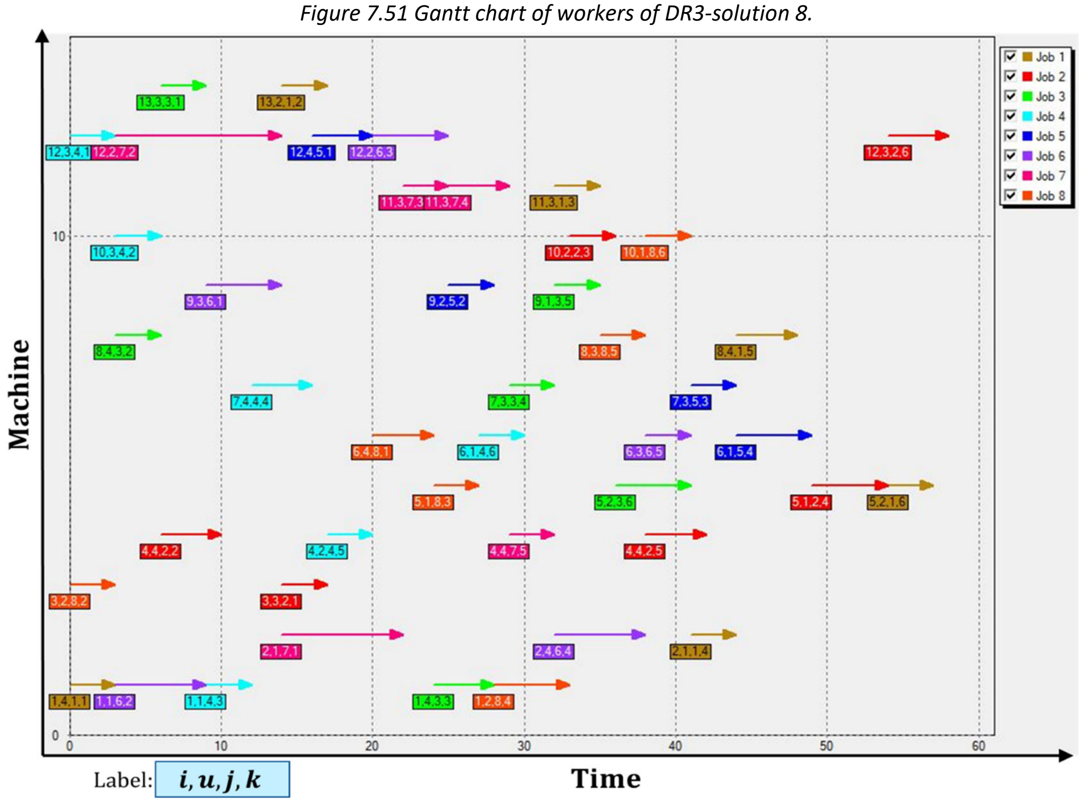


*Figure 7.52 Gantt chart of machines of DR3-solution 9.*

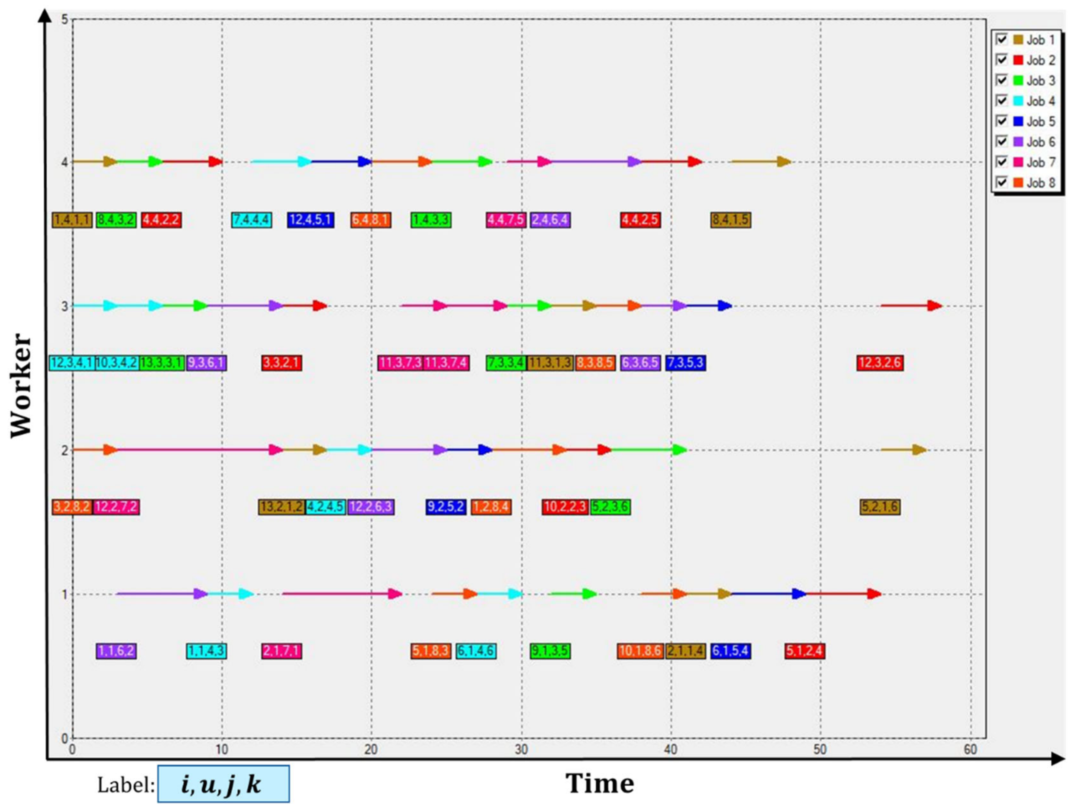



*Figure 7.53 Gantt chart of workers of DR3-solution 9.*

Table 7.20 shows the comparison of the results provided by NSGA-II and MRSLS for instance DR4. It can be noted that the NSGA-II generated a better result than the MRSLS by presenting an efficient front of 8 solutions. Gantt charts of the machines and workers for the Pareto-optimal solutions of the NSGA-II for DR4 are shown in Figure 7.54 to Figure 7.71.

*Table 7.20 Comparison of the NSGA-II and the MRSLS (DR4).*

| DR4 | NSGA-II | | | MRSLS | | |
|---|---|---|---|---|---|---|
| Solution | $c_{max}$ | $\sum_{j=1}^{n} W_j T_j$ | $L_{max}$ | $c_{max}$ | $\sum_{j=1}^{n} W_j T_j$ | $L_{max}$ |
| 1 | 67 | 1051 | 61 | 78 | 1404 | 71 |
| 2 | 73 | 869 | 63 | 92 | 1400 | 69 |
| 3 | 72 | 925 | 61 | 93 | 1417 | 68 |
| 4 | 76 | 909 | 61 | 77 | 1502 | 65 |
| 5 | 71 | 944 | 63 | | | |
| 6 | 64 | 1216 | 61 | | | |
| 7 | 64 | 1018 | 63 | | | |
| 8 | 65 | 1090 | 61 | | | |
| 9 | 68 | 990 | 61 | | | |
| Average time | 26.99 s | | | | | |

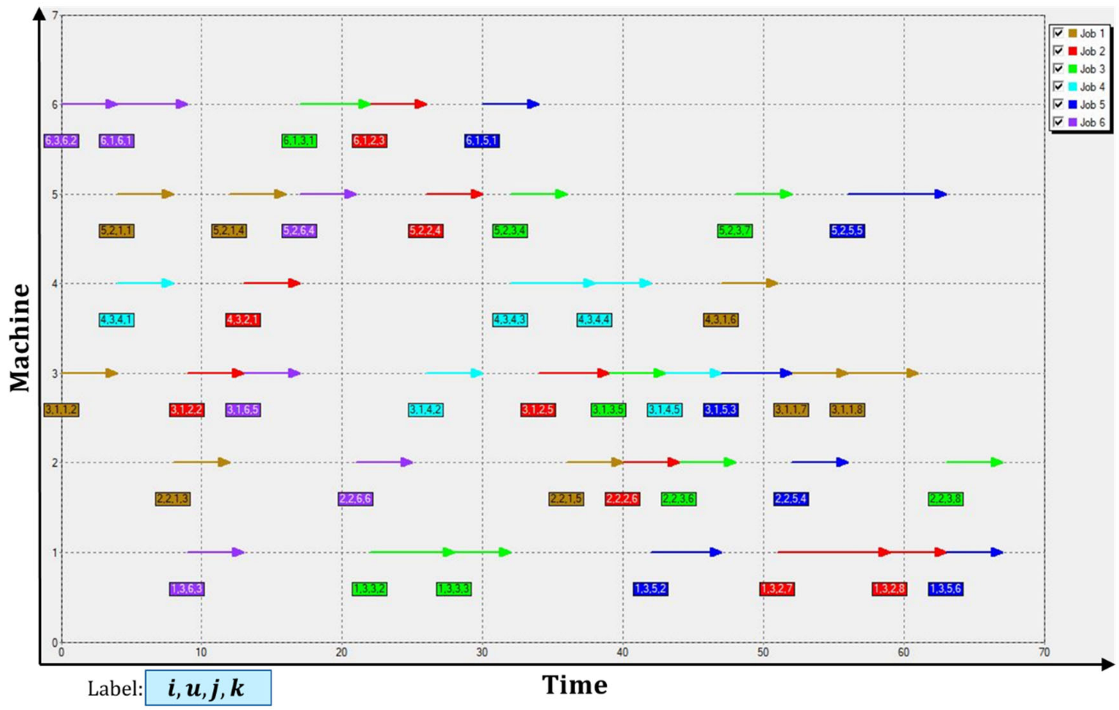



*Figure 7.54 Gantt chart of machines of DR4-solution 1.*

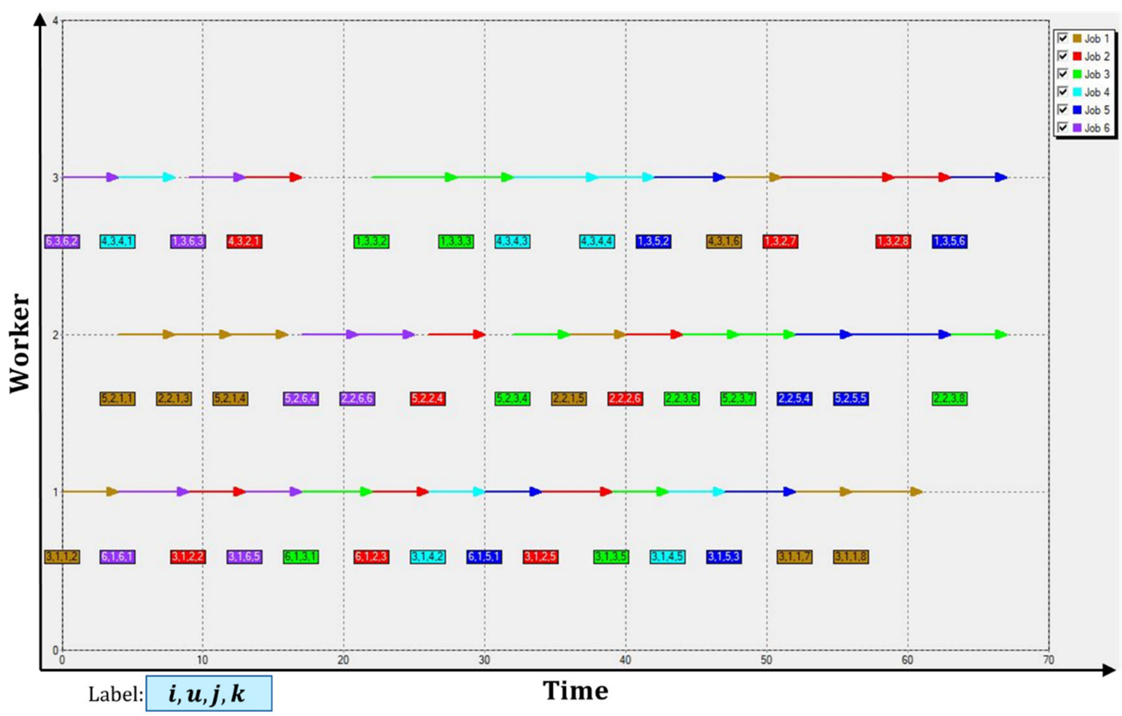



*Figure 7.55 Gantt chart of workers of DR4-solution 1.*

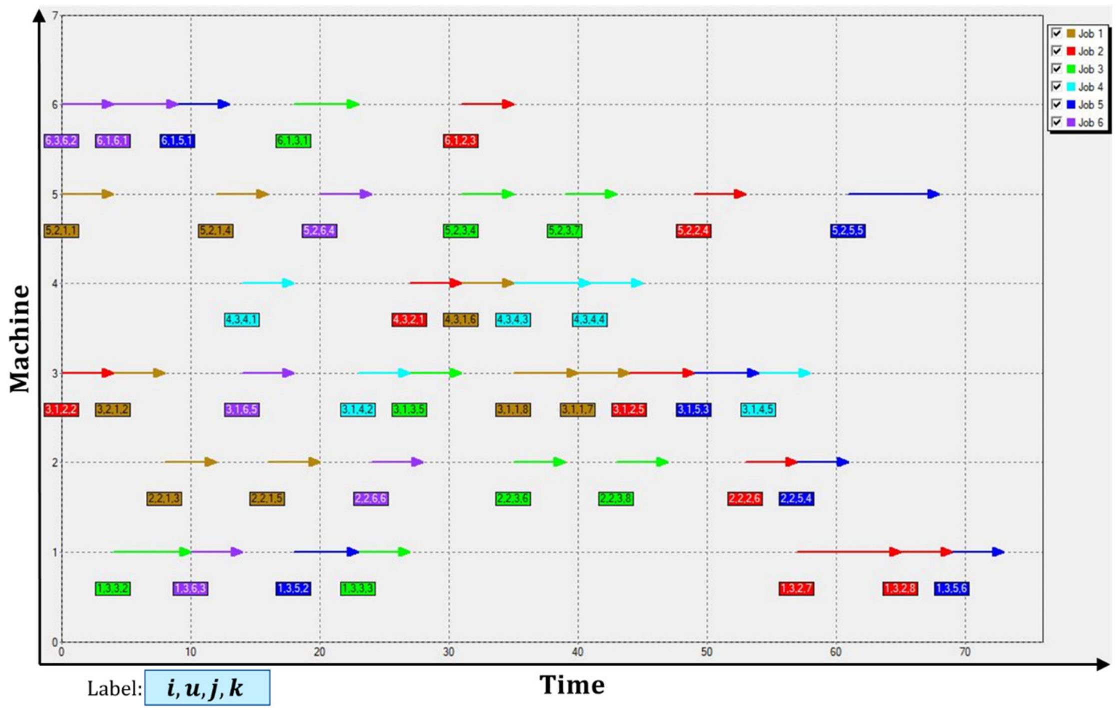



*Figure 7.56 Gantt chart of machines of DR4-solution 2.*

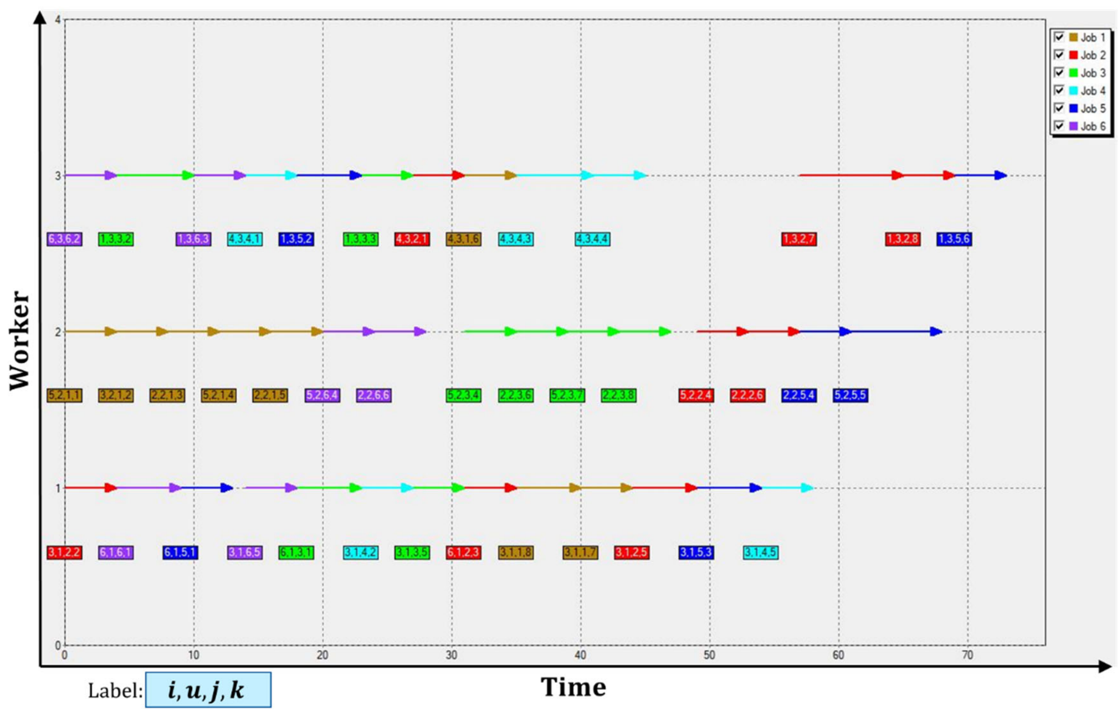



*Figure 7.57 Gantt chart of workers of DR4-solution 2.*

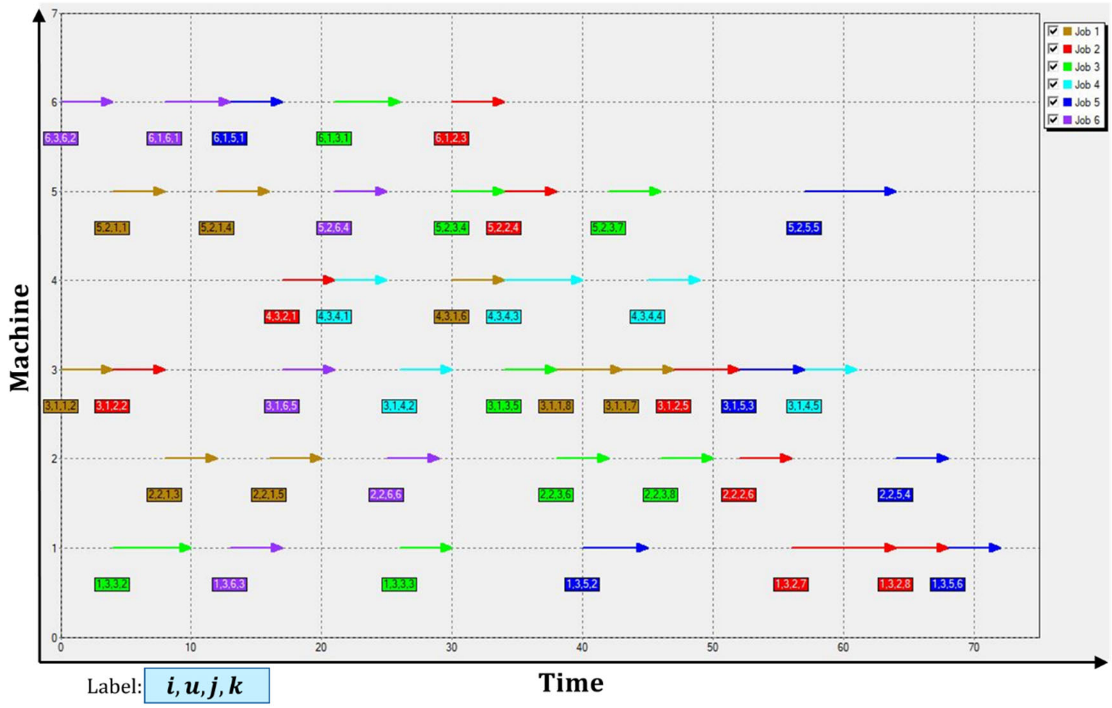



*Figure 7.58 Gantt chart of machines of DR4-solution 3.*

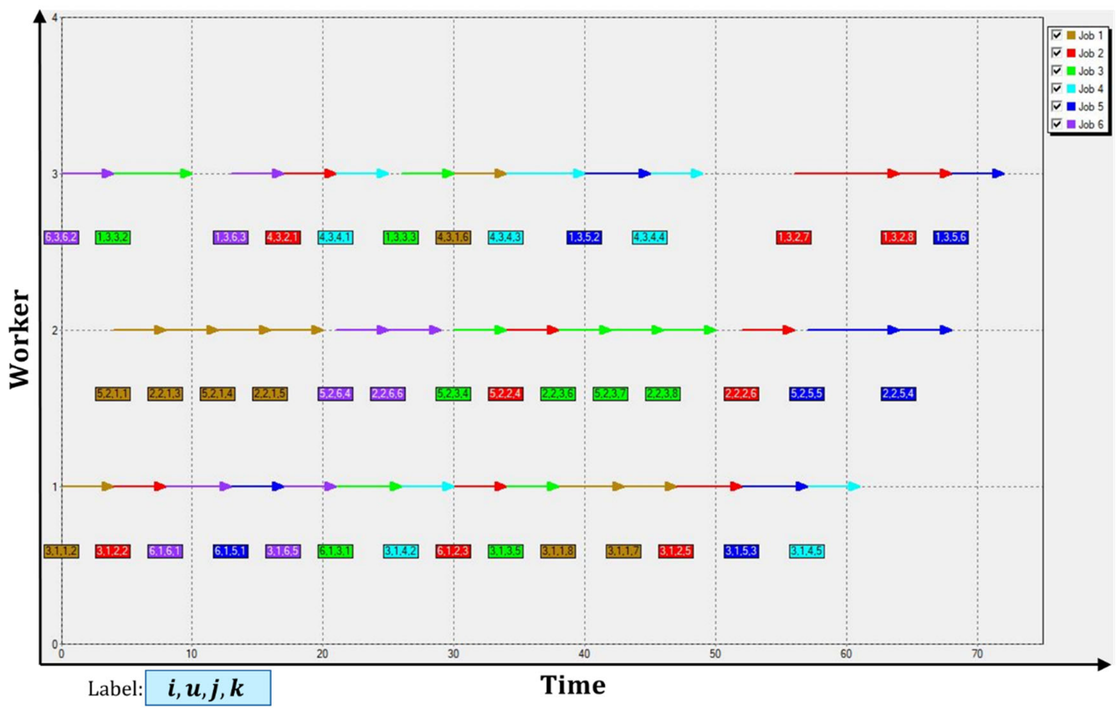



*Figure 7.59 Gantt chart of workers of DR4-solution 3.*

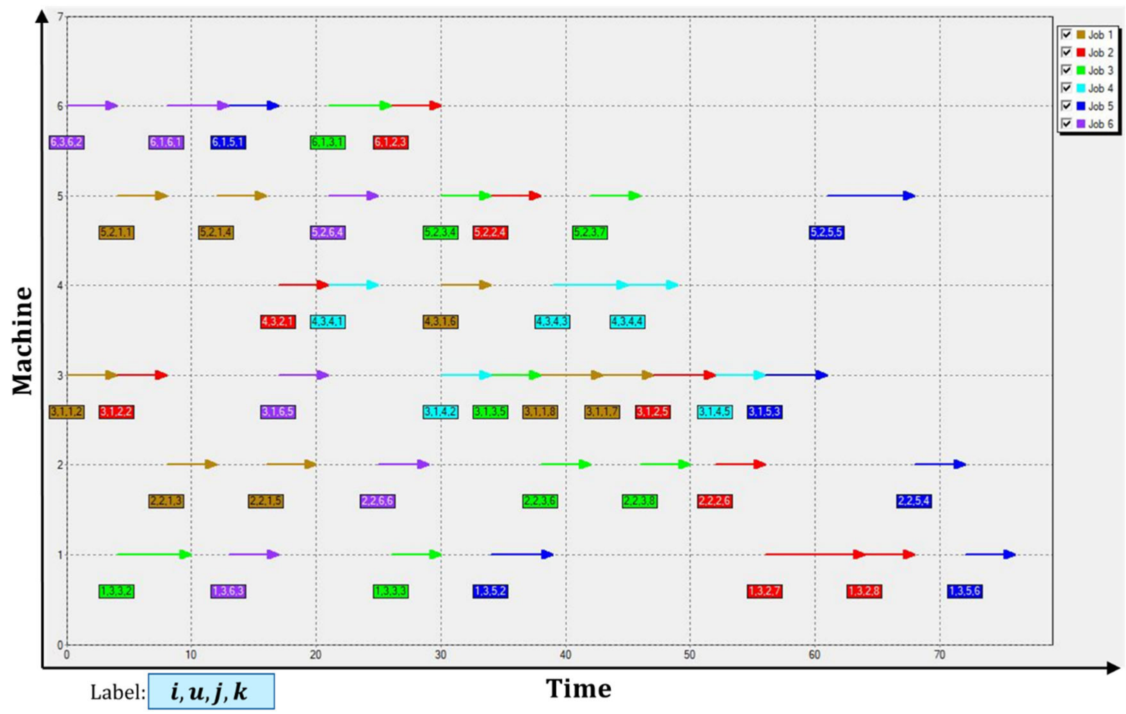



*Figure 7.60 Gantt chart of machines of DR4-solution 4.*

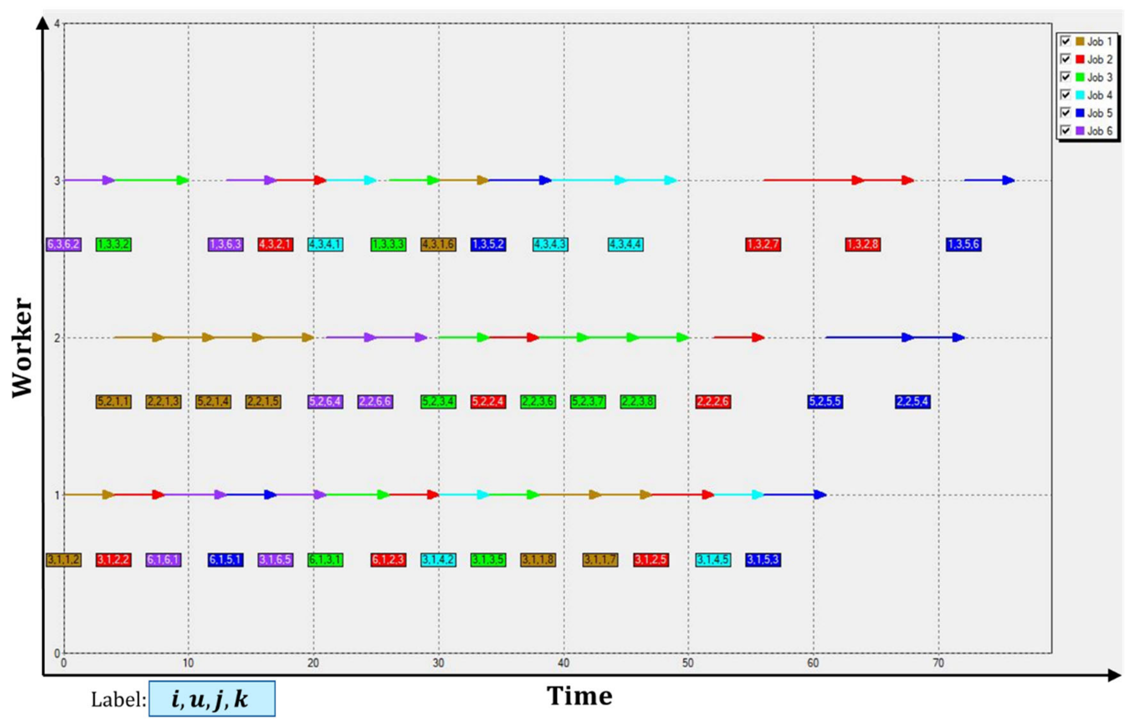



*Figure 7.61 Gantt chart of workers of DR4-solution 4.*

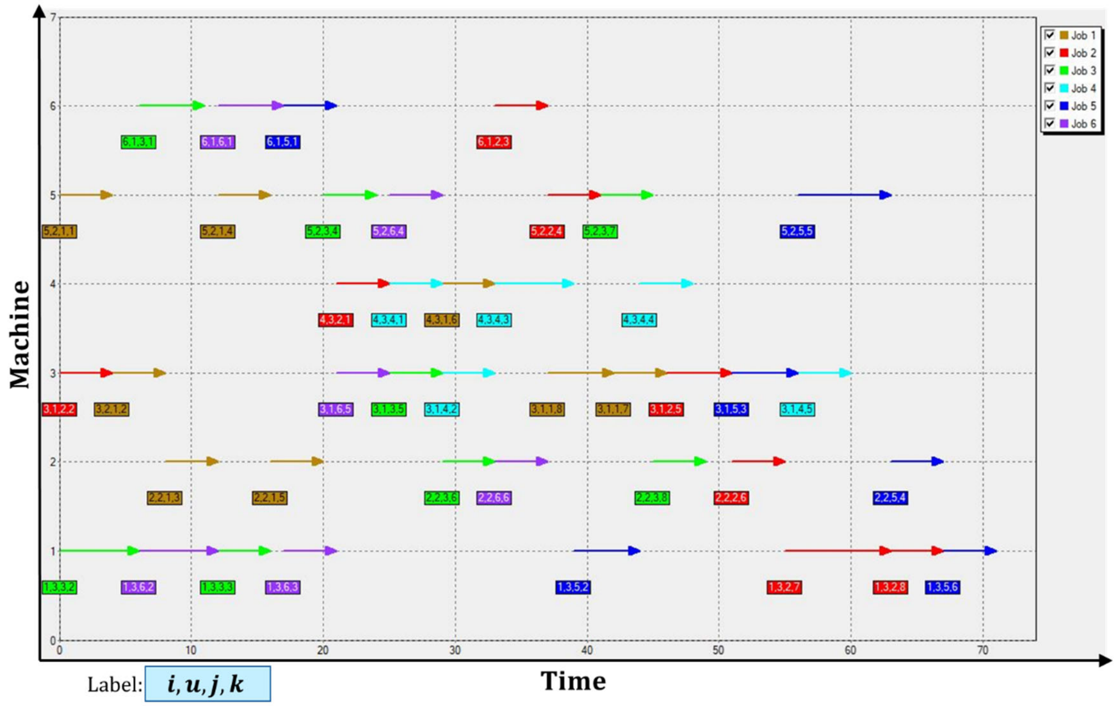



*Figure 7.62 Gantt chart of machines of DR4-solution 5.*

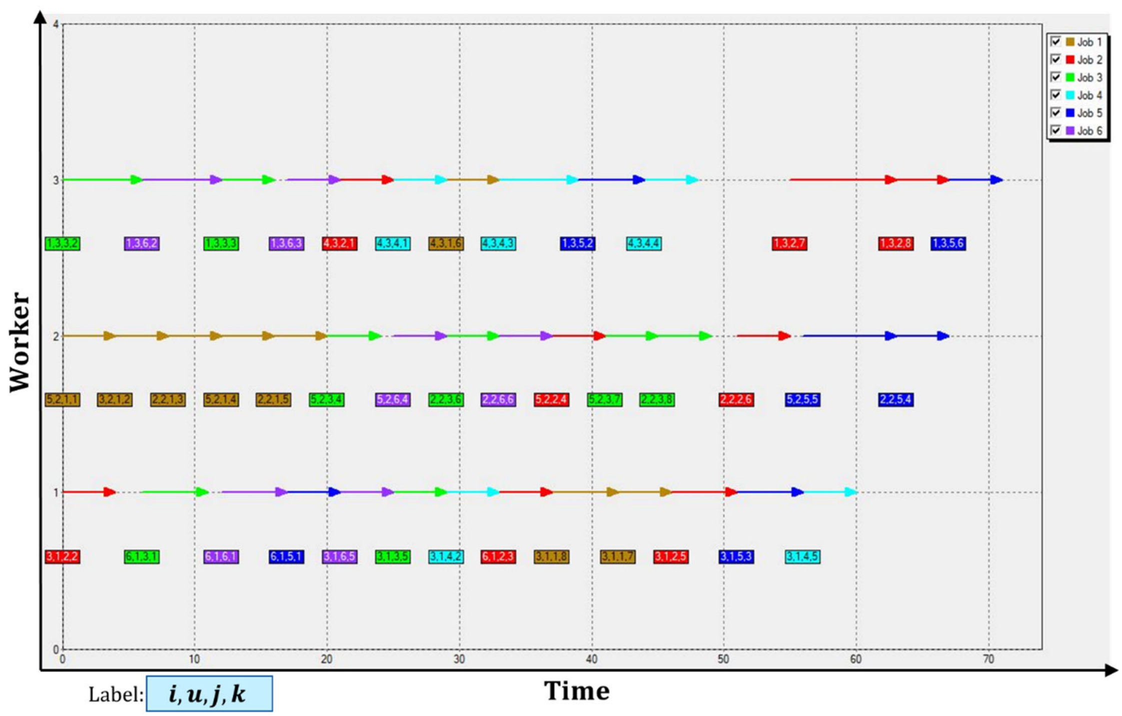



*Figure 7.63 Gantt chart of workers of DR4-solution 5.*

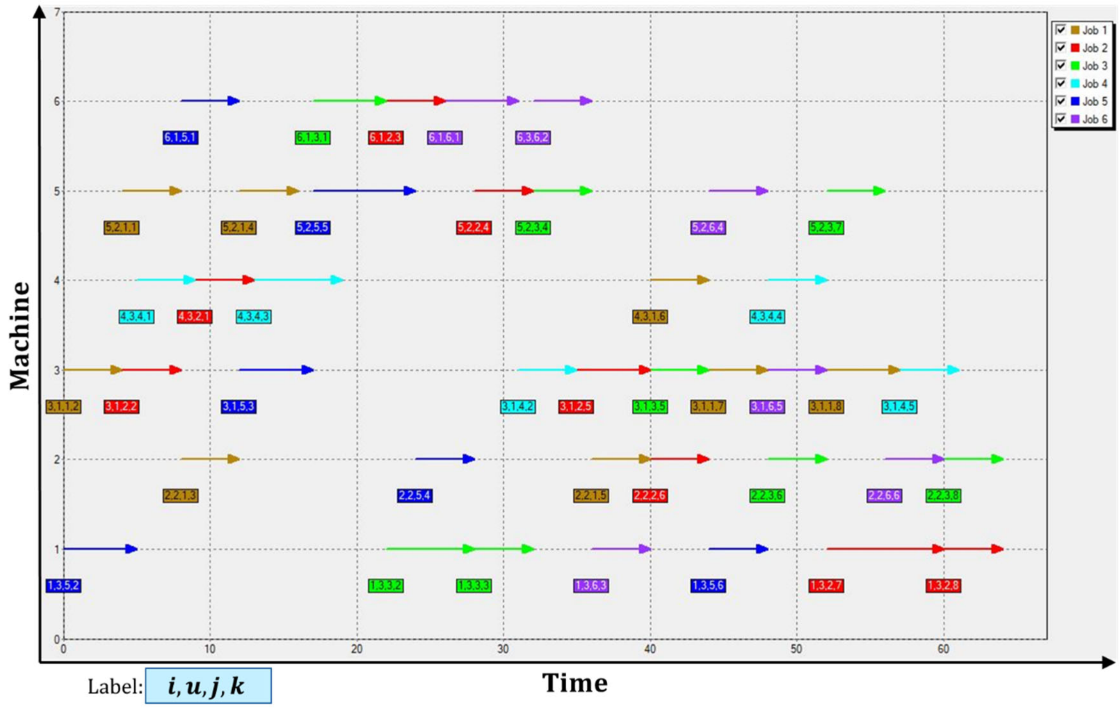



*Figure 7.64 Gantt chart of machines of DR4-solution 6.*

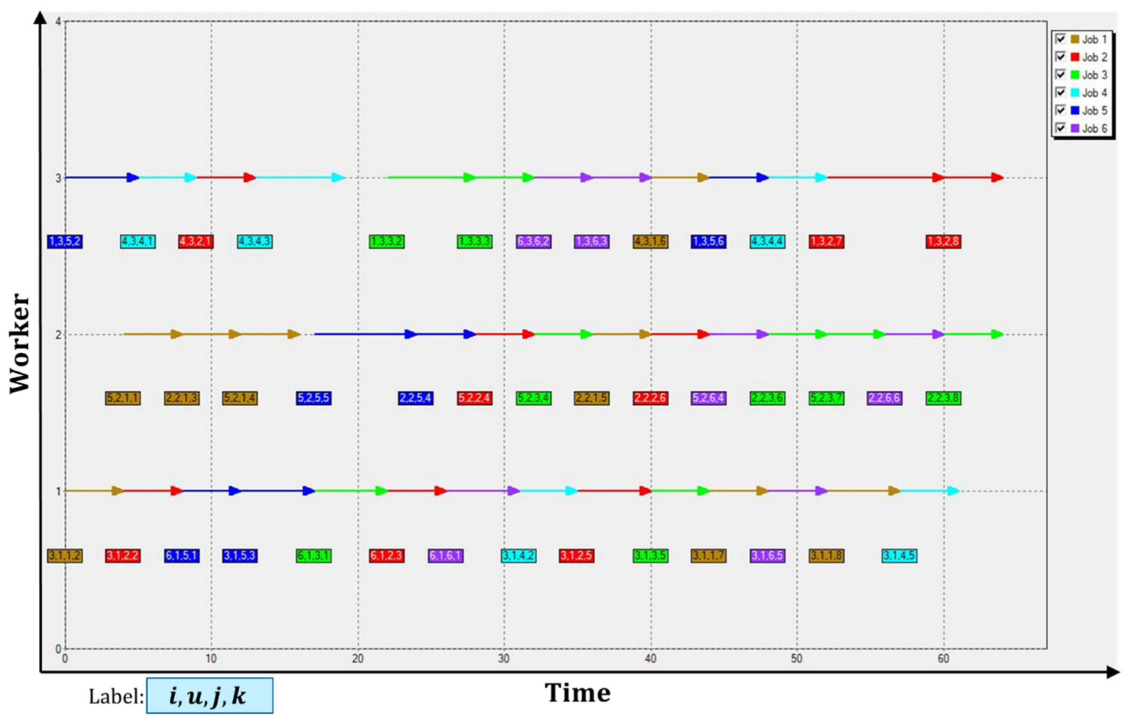



*Figure 7.65 Gantt chart of workers of DR4-solution 6.*

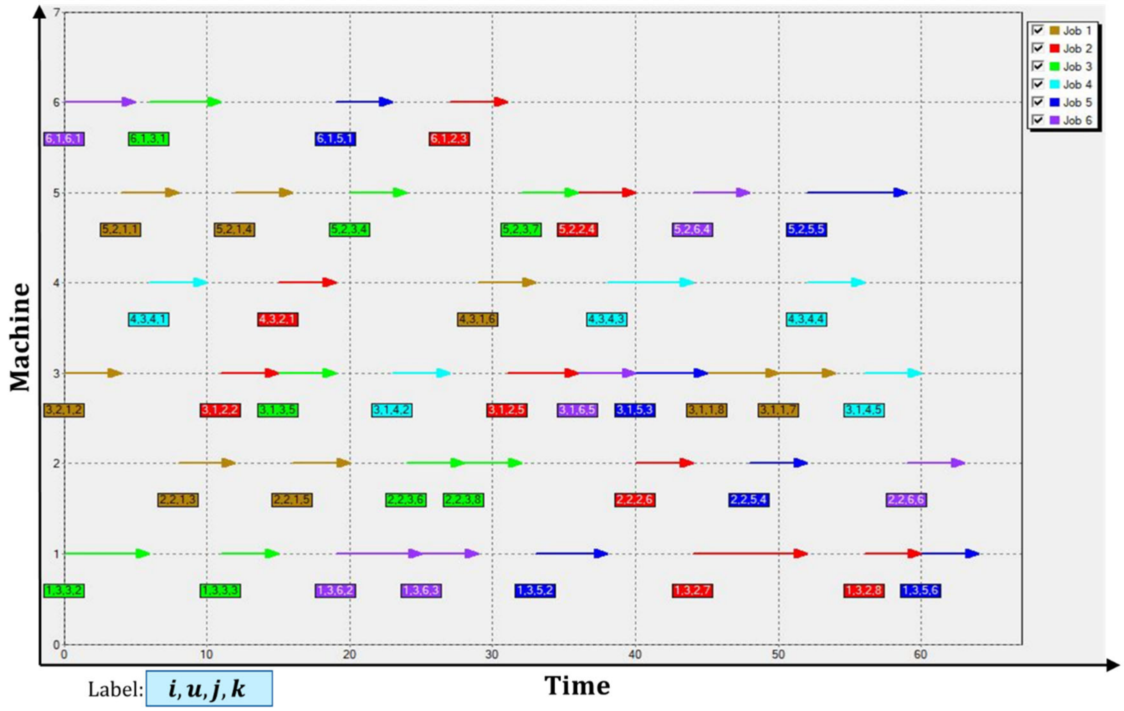



*Figure 7.66 Gantt chart of machines of DR4-solution 7.*

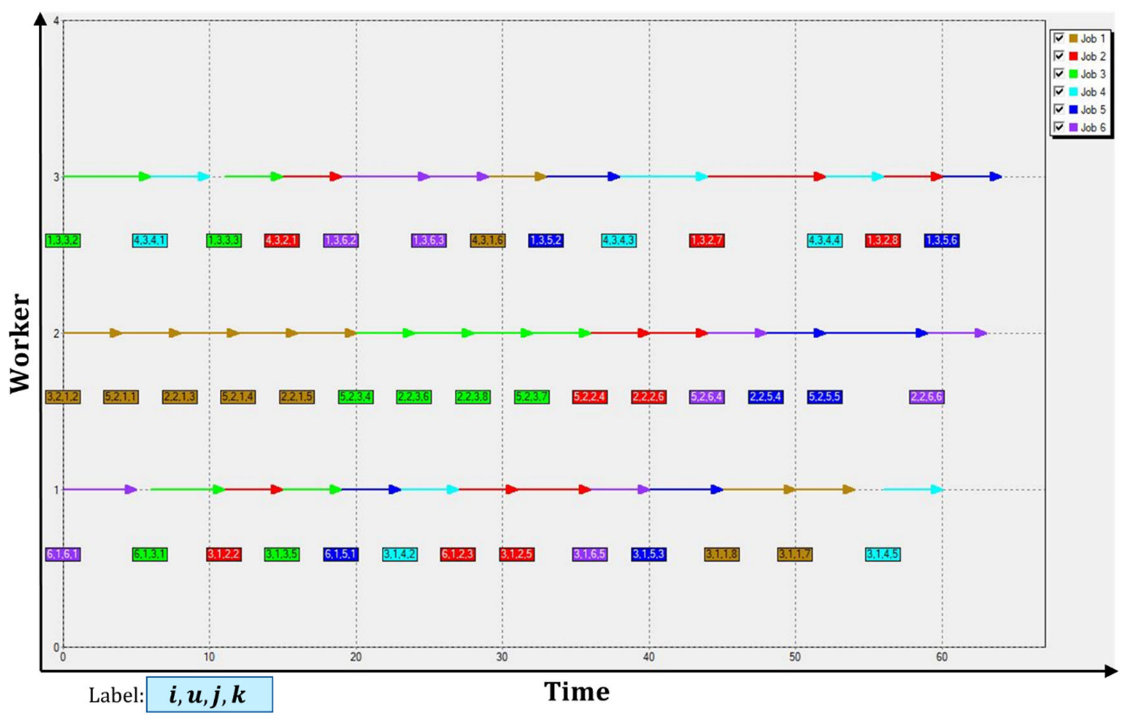



*Figure 7.67 Gantt chart of workers of DR4-solution 7.*

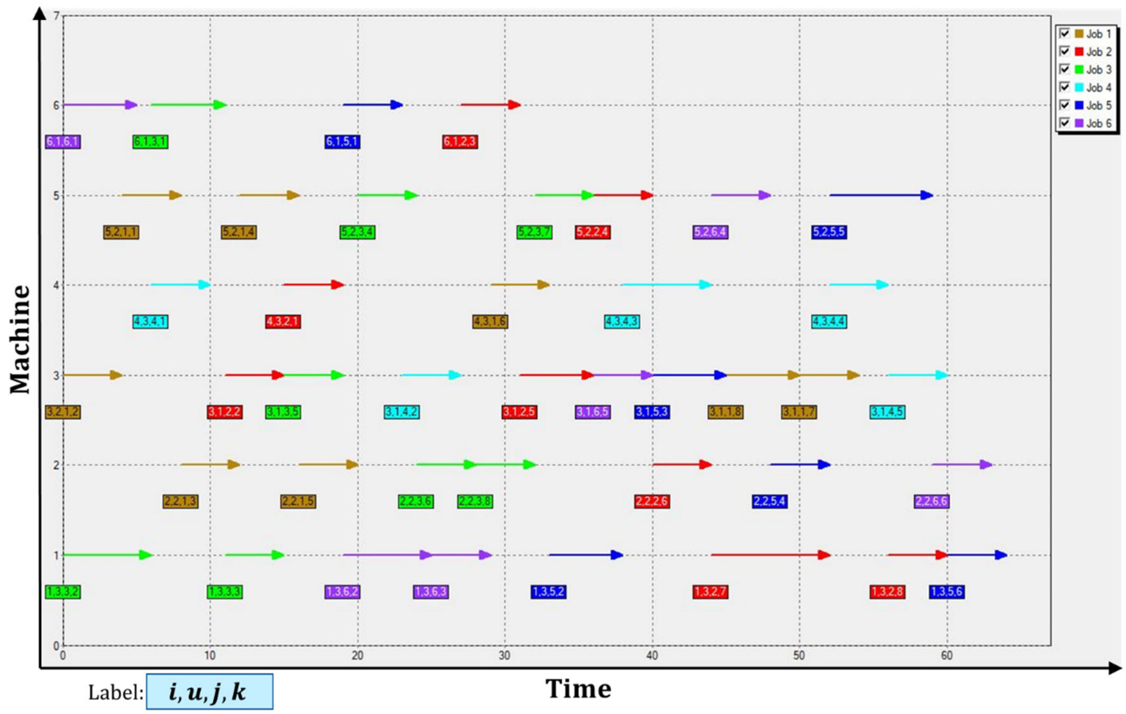



*Figure 7.68 Gantt chart of machines of DR4-solution 8.*

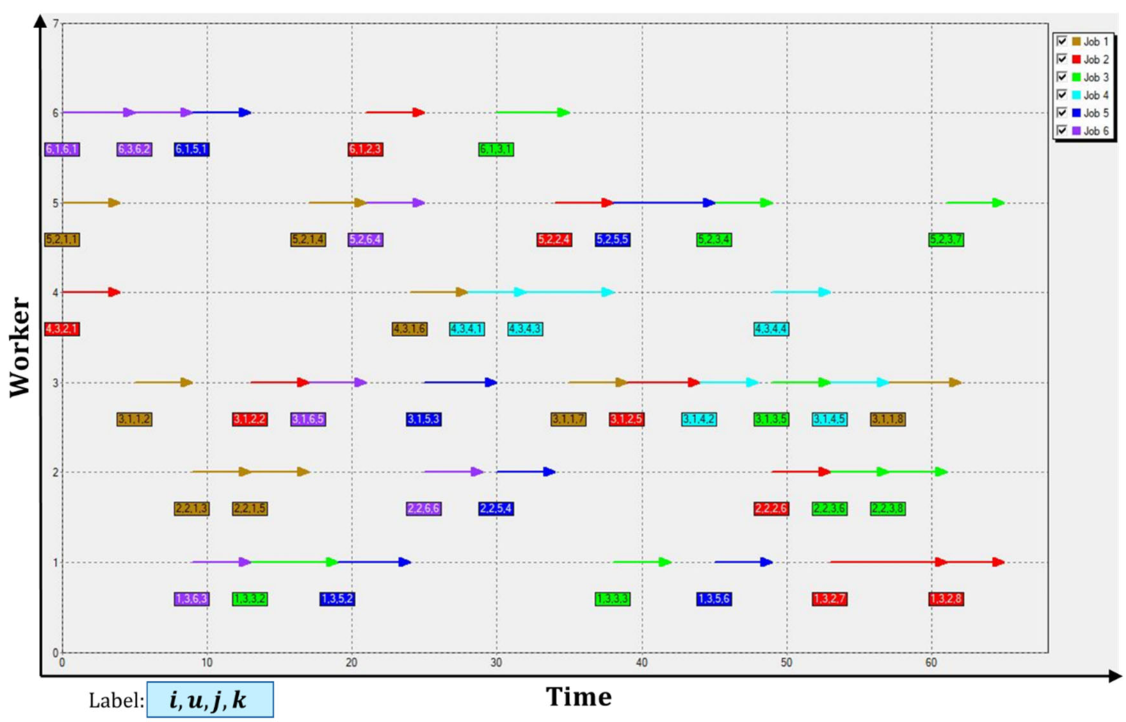



*Figure 7.69 Gantt chart of workers of DR4-solution 8.*

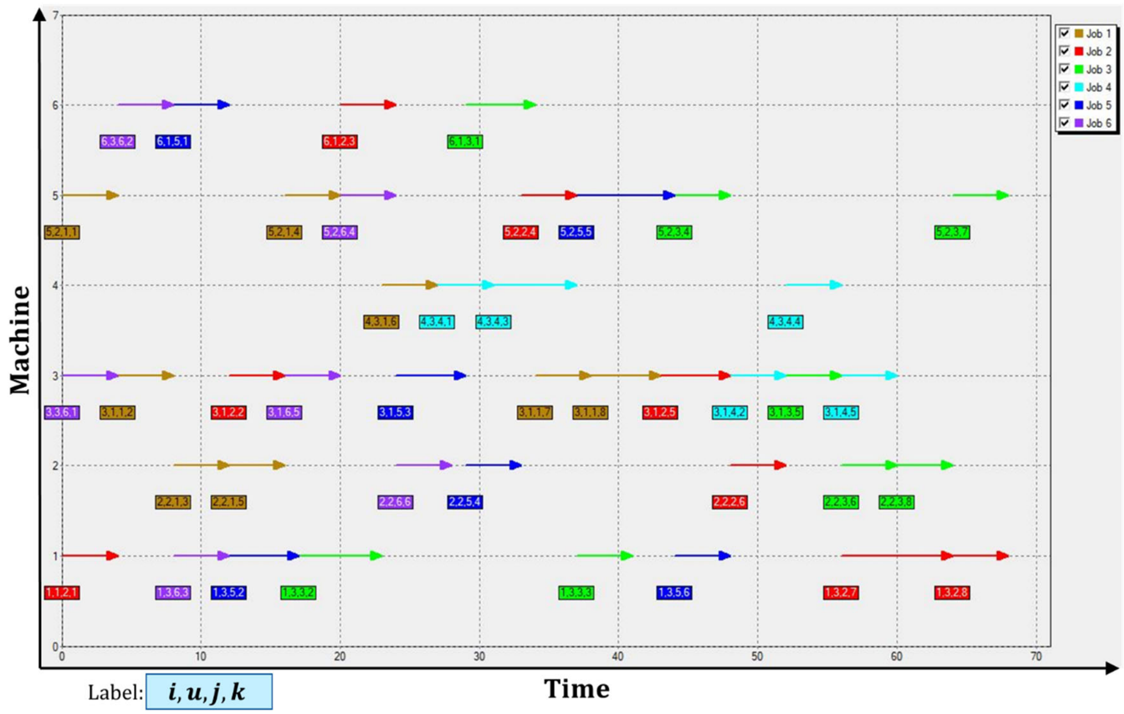



*Figure 7.70 Gantt chart of machines of DR4-solution 9.*

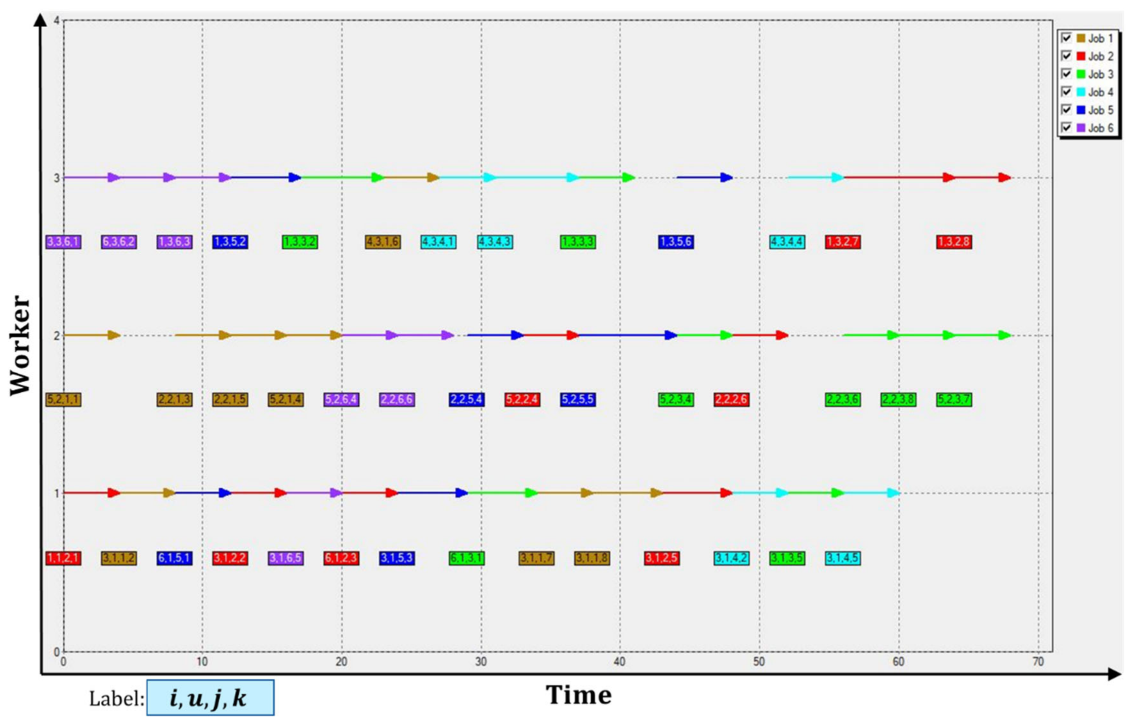



*Figure 7.71 Gantt chart of workers of DR4-solution 9.*

Table 7.21 shows the findings of NSGA-II and MRSLS. NSGA-II provided a total of 6 solutions that outperformed the solutions provide by MRSLS. Gantt charts of the machines and workers for the Pareto-optimal solutions of the NSGA-II for DR4 are shown in Figure 7.72 to Figure 7.83.

*Table 7.21 Comparison of the NSGA-II and the MRSLS (DR5).*

| DR5 | NSGA-II | | | MRLS | | |
|---|---|---|---|---|---|---|
| Solution | $c_{max}$ | $\sum_{j=1}^{n} W_j T_j$ | $L_{max}$ | $c_{max}$ | $\sum_{j=1}^{n} W_j T_j$ | $L_{max}$ |
| 1 | 43 | 546 | 32 | 69 | 1114 | 61 |
| 2 | 41 | 531 | 34 | 68 | 1346 | 47 |
| 3 | 50 | 498 | 32 | | | |
| 4 | 44 | 467 | 34 | | | |
| 5 | 43 | 503 | 34 | | | |
| 6 | 42 | 527 | 34 | | | |
| Average time | 40.8 s | | | | | |

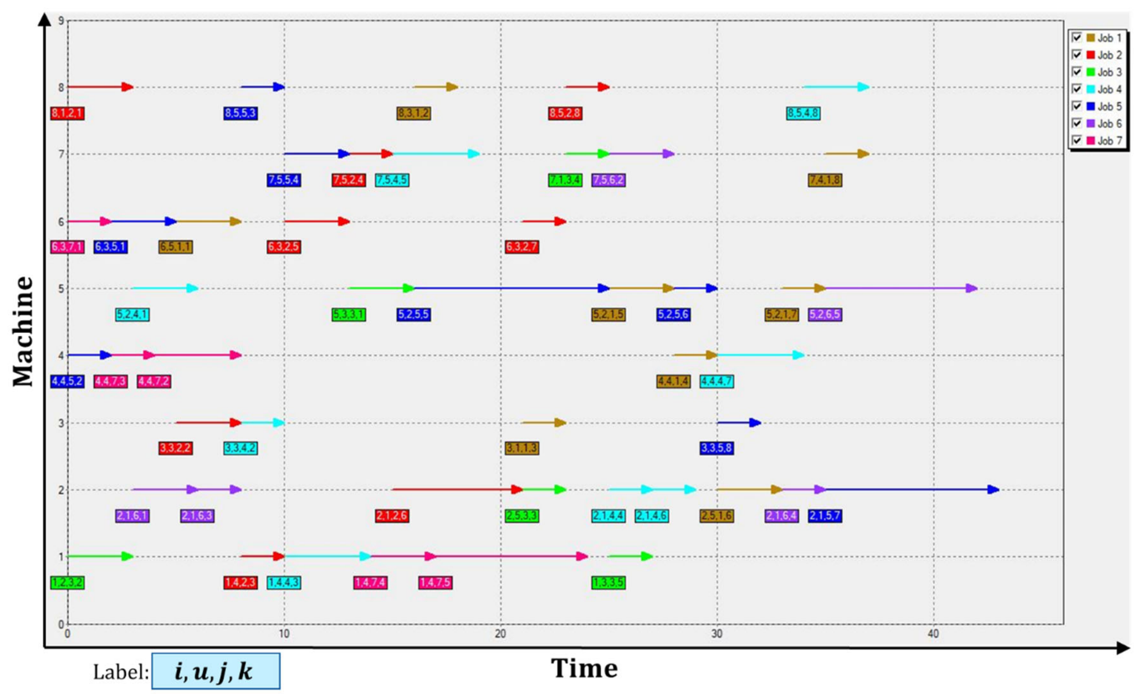



*Figure 7.72 Gantt chart of machines of DR5-solution 1.*

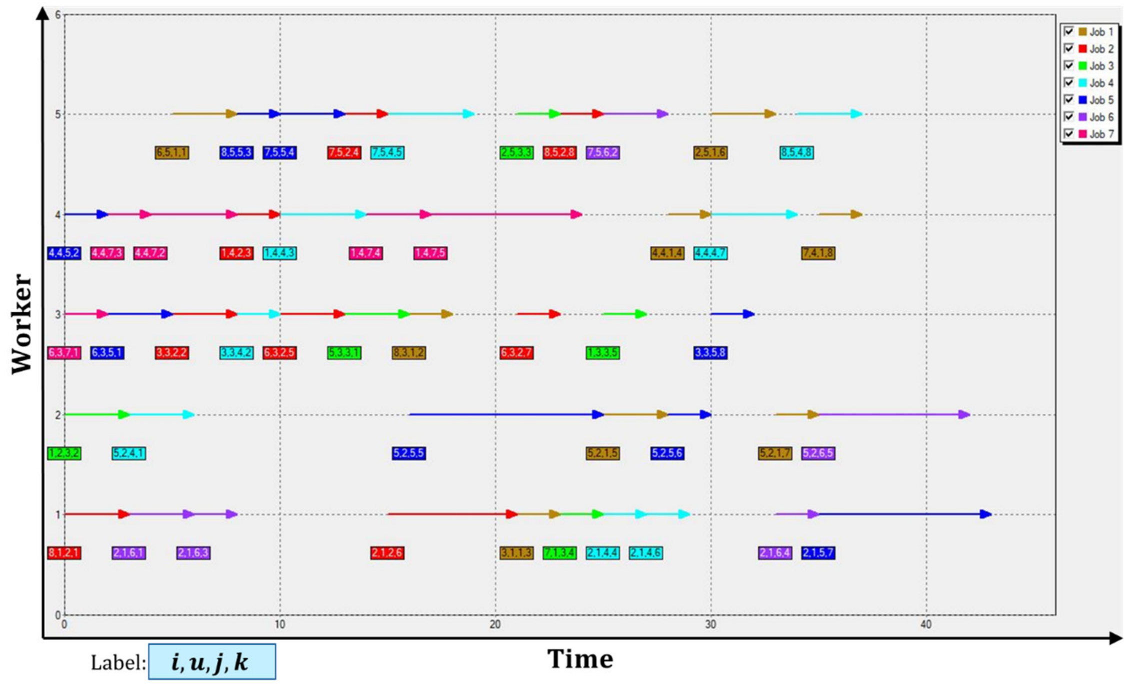



*Figure 7.73 Gantt chart of workers of DR5-solution 1.*

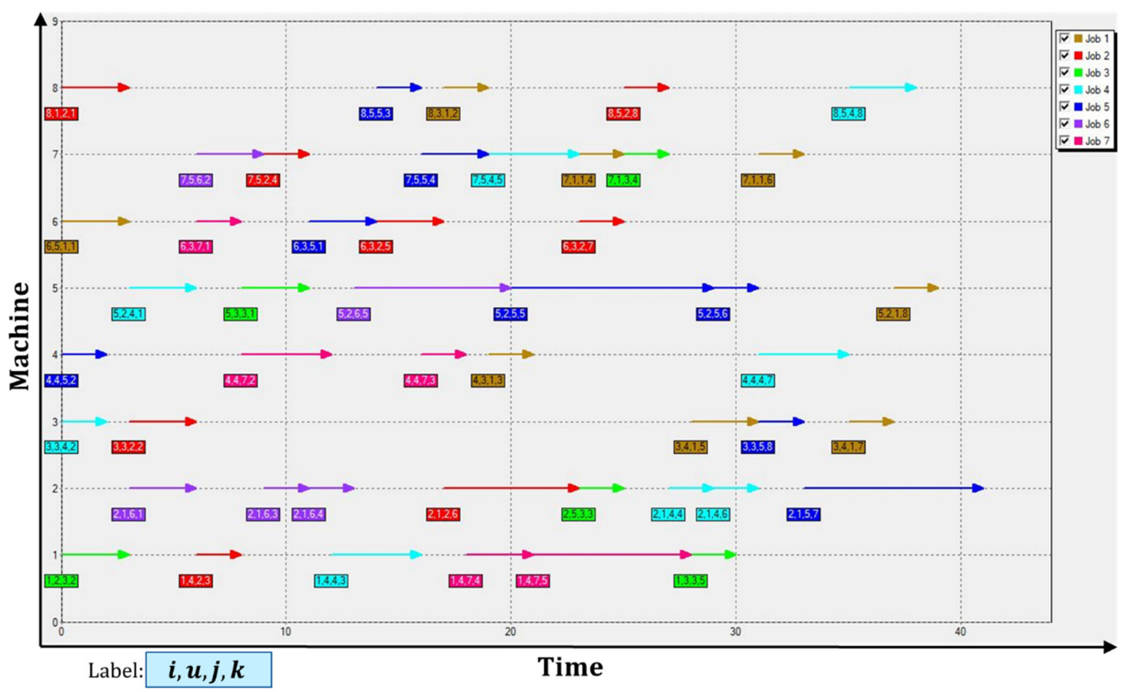



*Figure 7.74 Gantt chart of machines of DR5-solution 2.*

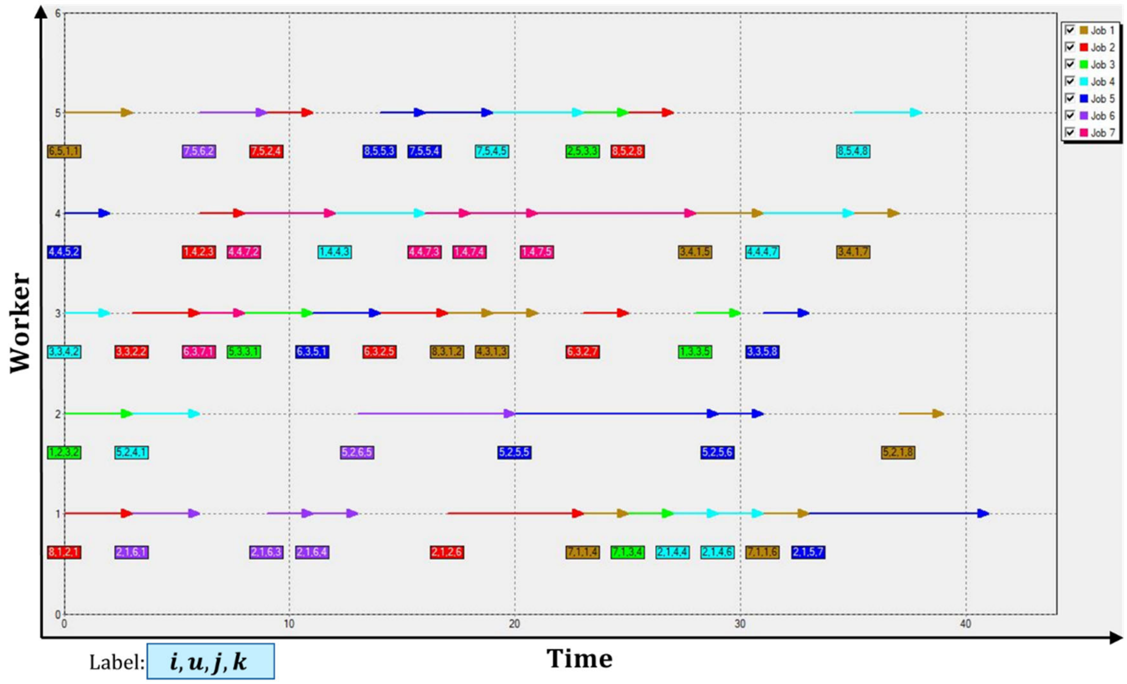



*Figure 7.75 Gantt chart of workers of DR5-solution 2.*

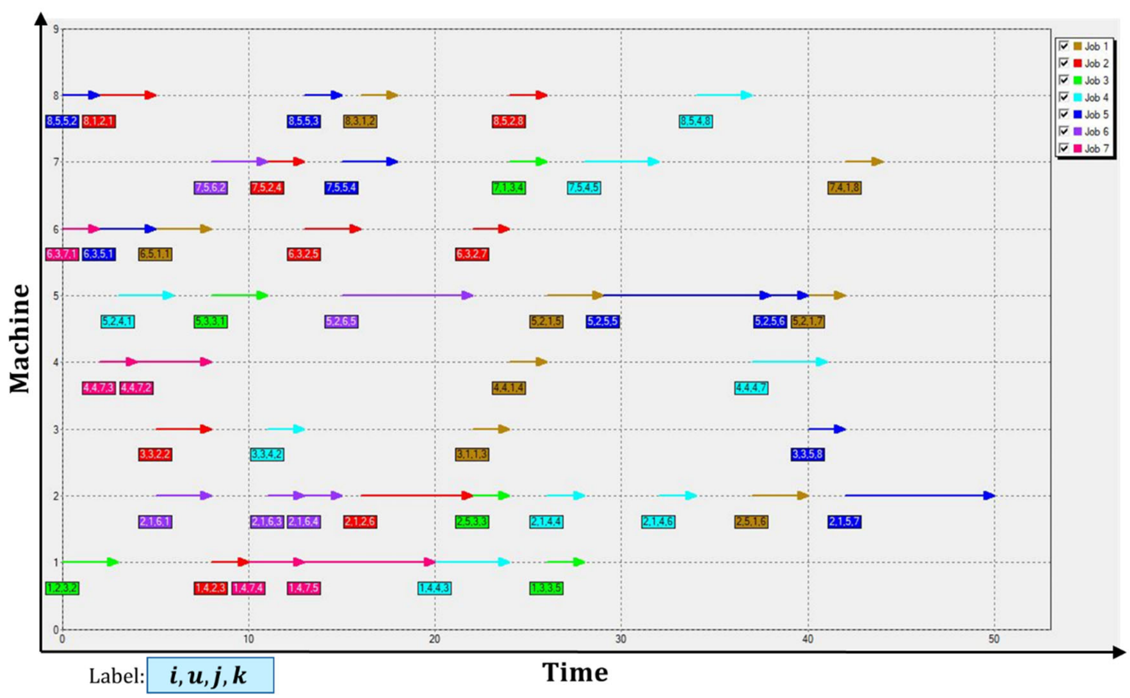



*Figure 7.76 Gantt chart of machines of DR5-solution 3.*

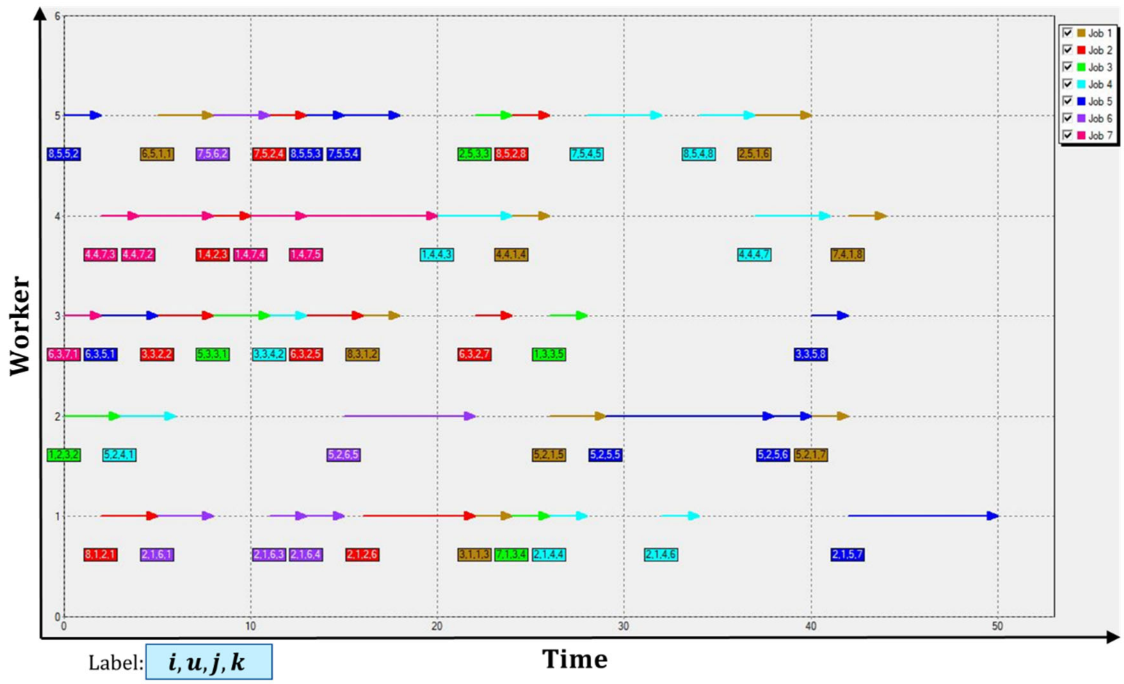



*Figure 7.77 Gantt chart of workers of DR5-solution 3.*

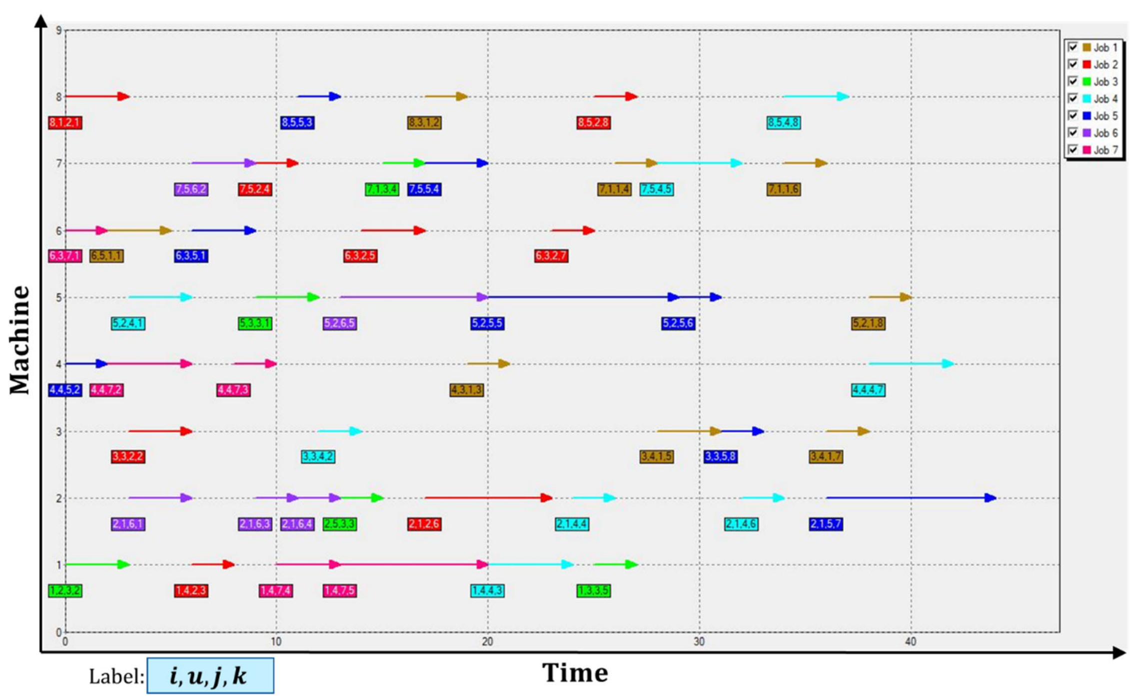



*Figure 7.78 Gantt chart of machines of DR5-solution 4.*

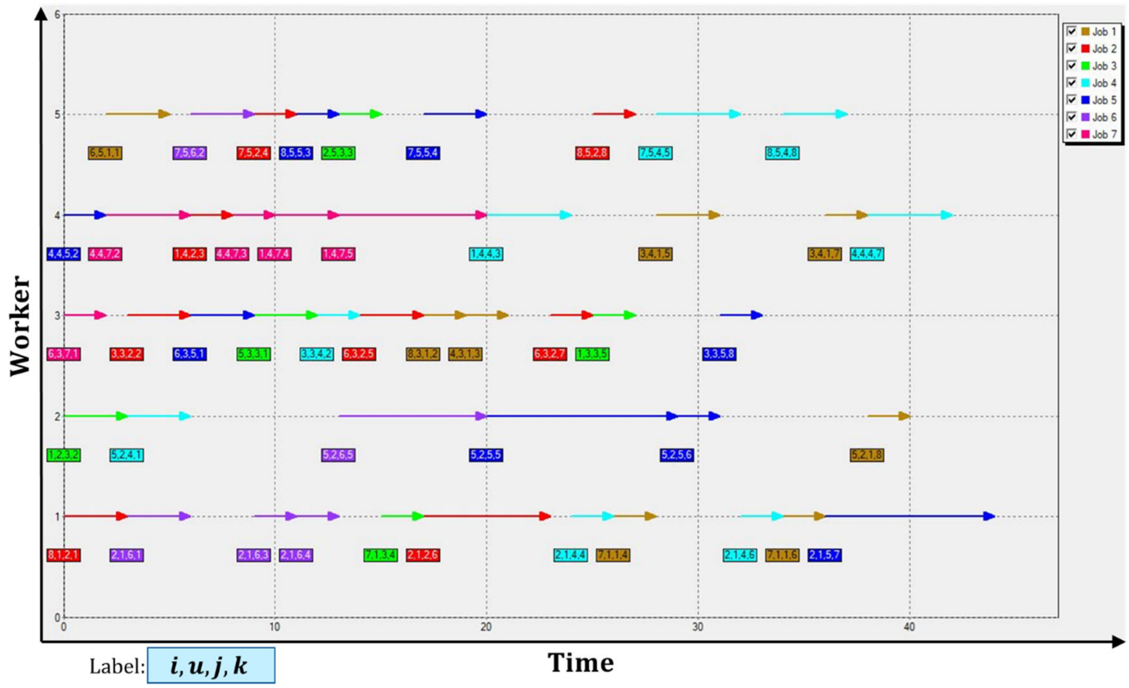



*Figure 7.79 Gantt chart of workers of DR5-solution 4.*

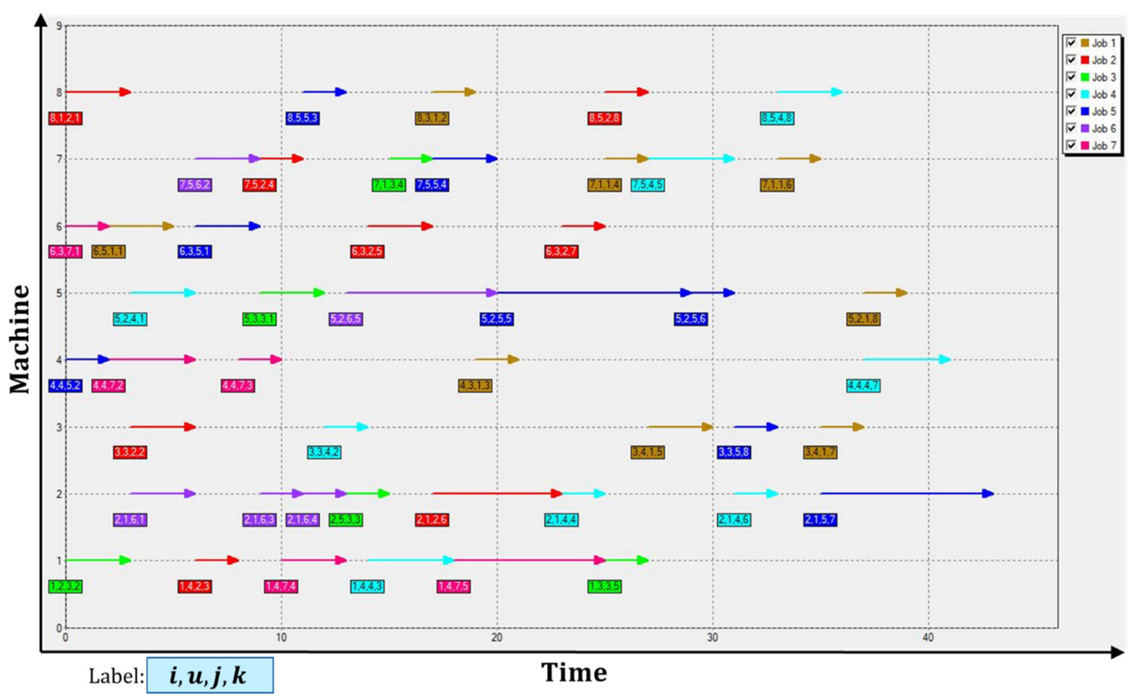



*Figure 7.80 Gantt chart of machines of DR5-solution 5.*

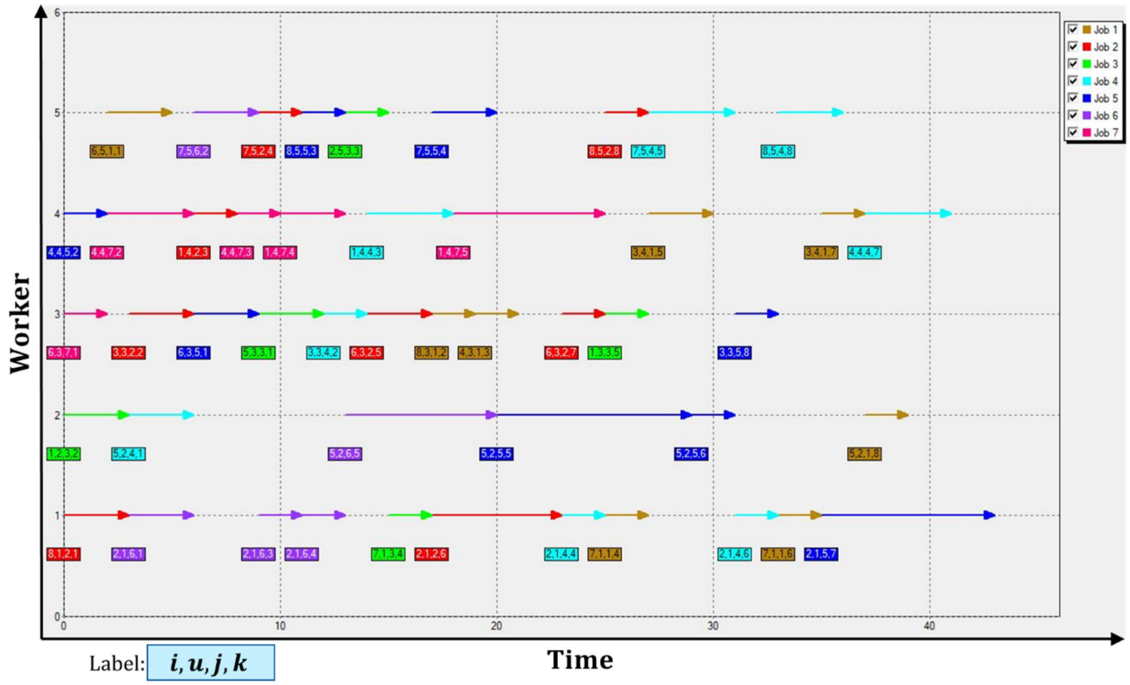



*Figure 7.81 Gantt chart of workers of DR5-solution 5.*

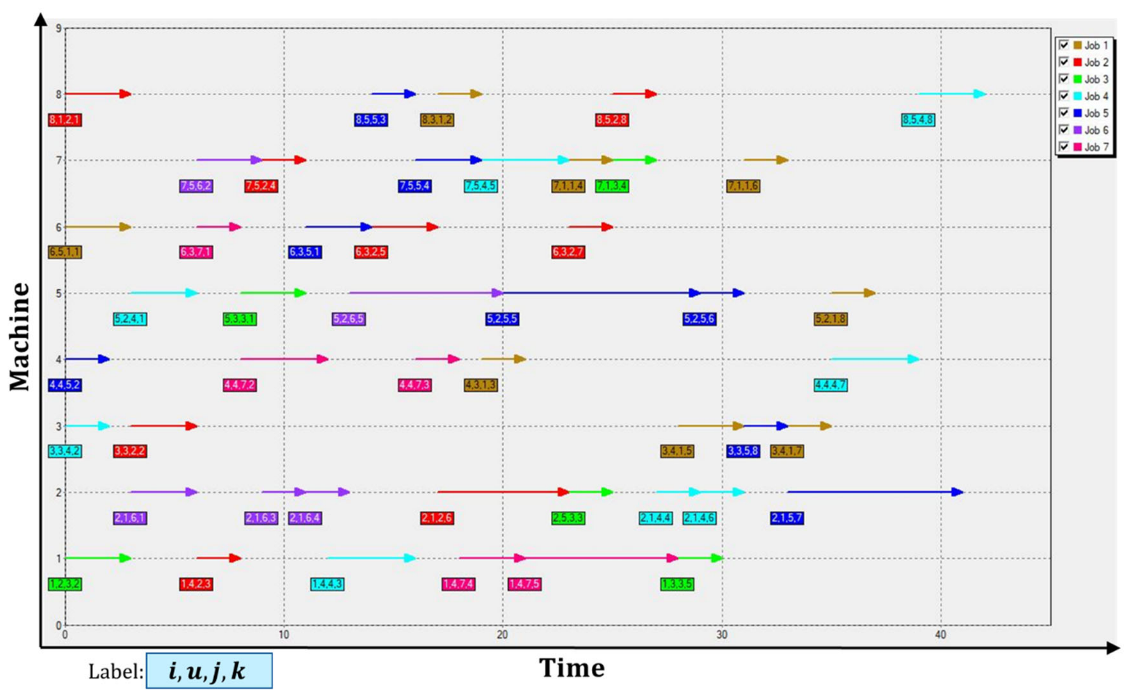



*Figure 7.82 Gantt chart of machines of DR5-solution 6.*

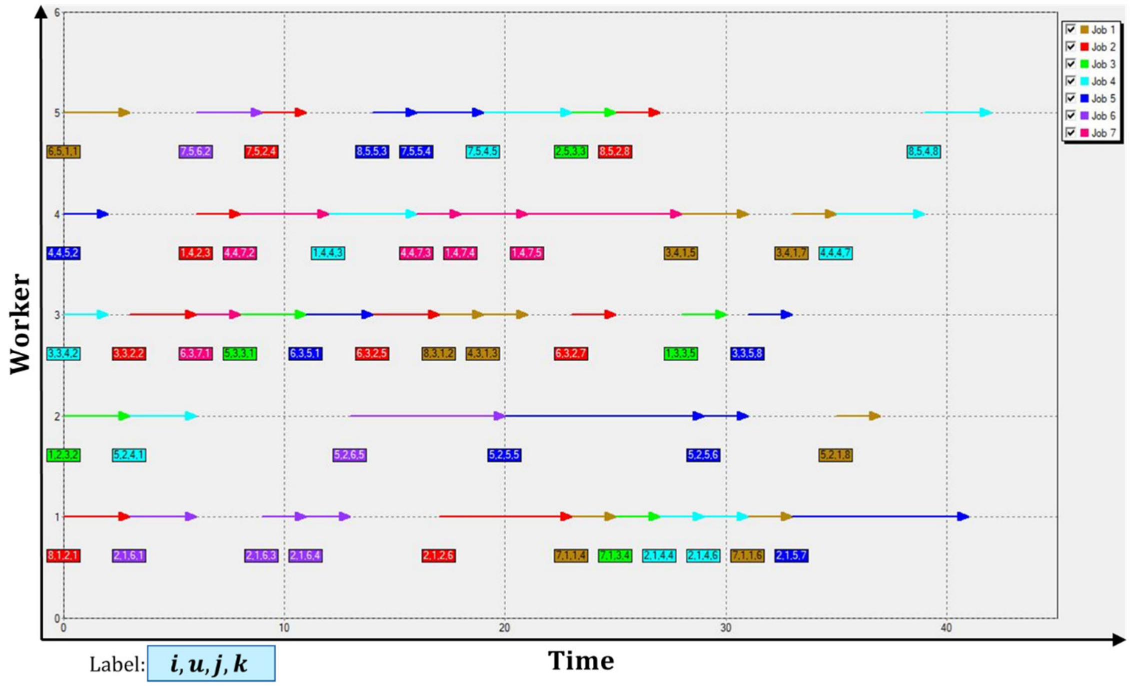



*Figure 7.83 Gantt chart of workers of DR5-solution 6.*

In order to evaluate the performance of the NSGA-II and MRSLS, the solutions were sorted to detect the percentage of solutions that contribute to the non-dominated set. Table 7.22 shows that NSGA-II contributes 100% to the non-dominated set for all the instances. This result shows that NSGA-II outperformed MRSLS.

*Table 7.22 Percentage of contribution to the non-dominated set.*

| Instance | %(NSGA-II) | %( MRSLS) |
|---|---|---|
| DR1 | 100 | 0 |
| DR2 | 100 | 0 |
| DR3 | 100 | 0 |
| DR4 | 100 | 0 |
| DR5 | 100 | 0 |

## 7.4 Numerical experience for the FJSP-2F

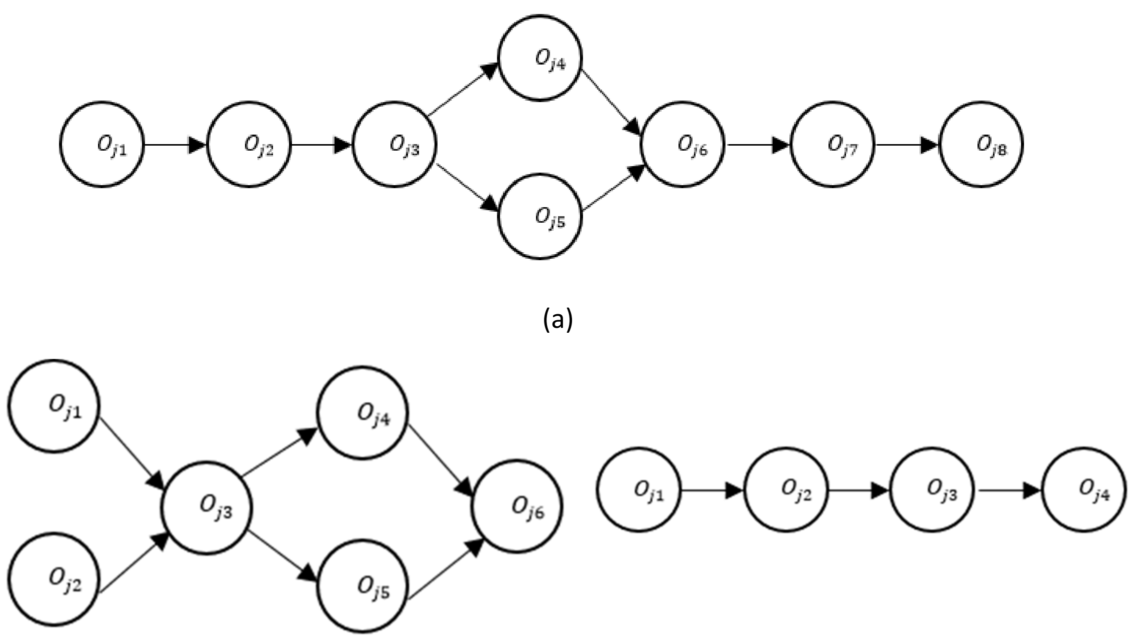
Three instances have been generated named PP1, PP2 and PP3. Processing times, processing cost, weights, due dates and the number of alternative process plans were generated from a uniform distribution, and the parameter are presented in Table 7.23. Precedence relationships graphs are presented in Figure 7.84 and Figure 7.85. Precedence tables are shown in Table 7.24., Table 7.25 and Table 7.26 respectively. The instances data are shown in APPENDIX A (Table A.29 to Table A.45).

Xpress Optimizer 25.01.05 algebraic model language and optimizer have been used to implement and solve the mathematical model for the FJSP-2F. The experiment was performed in PC with a processor Intel i3-6100 2.3GHz with 8GB in RAM. The weighted sum approached has been considered with Eq. (5.23). Four sets of weights are randomly generated for each instance. The weights are shown in Table 7.27 to Table 7.29.

.

*Table 7.23 Parameter for instances PP1, PP2 and PP3.*

| Name | *n x m* | ope | $P_{ijkp}$ | $g_{ijkp}$ | $d_j$ | $W_j$ | $b_j$ |
|---|---|---|---|---|---|---|---|
| PP1 | 10 x 5 | 146 | 5-15 | 1-18 | 10-25 | 1-10 | 2-4 |
| PP2 | 5 x 8 | 70 | 2-10 | 5-15 | 6-20 | 1-10 | 2-3 |
| PP3 | 6 x 6 | 11 | 4-8 | 10-20 | 10-20 | 1-10 | 2-4 |




(a)

(b)

(c)

*Figure 7.84 Operation precedence graphs.*

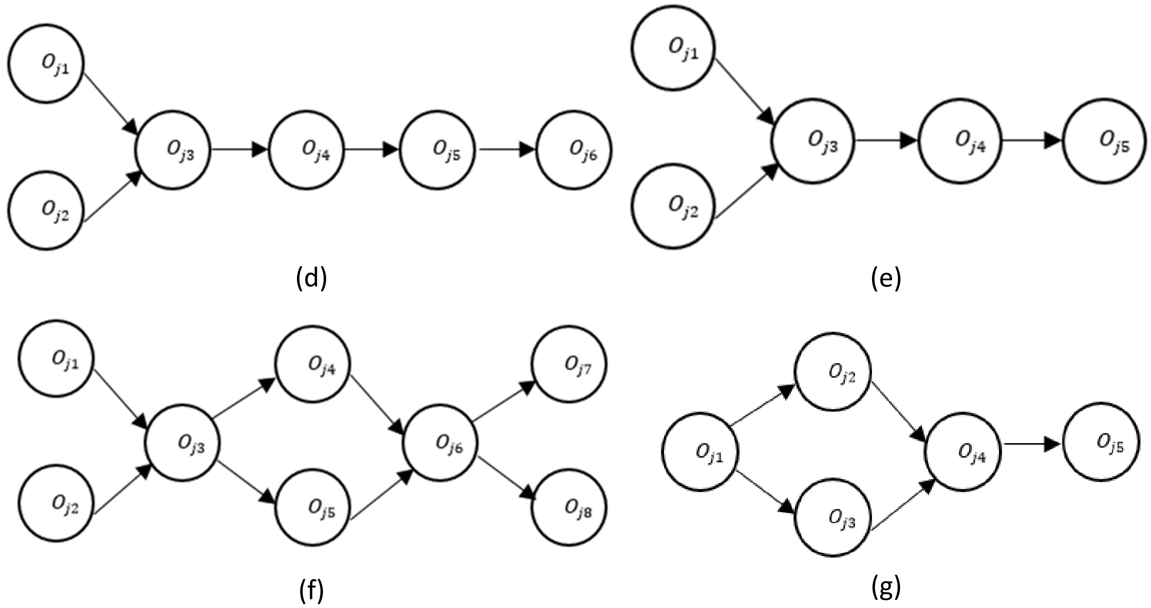



*Figure 7.85 Operation precedence graphs continued.*

*Table 7.24 Operation precedence graph PP1.*

| job | $b_j$ | $p$ | Operation precedence graph ( Figure 7.84 and Figure 7.85) |
|---|---|---|---|
| 1 | 2 | 1 | (e) |
| | | 2 | (c) |
| 2 | 3 | 1 | (b) |
| | | 2 | (d) |
| | | 3 | (g) |
| 3 | 2 | 1 | (a) |
| | | 2 | (f) |
| 4 | 3 | 1 | (c) |
| | | 2 | (e) |
| | | 3 | (g) |
| 5 | 2 | 1 | (d) |
| | | 2 | (g) |
| 6 | 4 | 1 | (b) |
| | | 2 | (d) |
| | | 3 | (e) |
| | | 4 | (g) |
| 7 | 2 | 1 | (a) |
| | | 2 | (f) |
| 8 | 2 | 1 | (e) |
| | | 2 | (c) |
| 9 | 3 | 1 | (b) |
| | | 2 | (d) |
| | | 3 | (g) |
| 10 | 2 | 1 | (a) |
| | | 2 | (f) |

*Table 7.25 Operation precedence graph PP2.*

| job | $b_j$ | $p$ | Operation precedence graph ( Figure 7.84 and Figure 7.85) |
|---|---|---|---|
| 1 | 3 | 1 | (h) |
| | | 2 | (a) |
| | | 3 | (f) |
| 2 | 2 | 1 | (e) |
| | | 2 | (g) |
| 3 | 2 | 1 | (b) |
| | | 2 | (d) |
| 4 | 3 | 1 | (e) |
| | | 2 | (g) |
| | | 3 | (c) |
| 5 | 2 | 1 | (d) |
| | | 2 | (g) |

*Table 7.26 Operation precedence graph PP1.*

| job | $b_j$ | $p$ | Operation precedence graph ( Figure 7.84 and Figure 7.85) |
|---|---|---|---|
| 1 | 2 | 1 | (b) |
| | | 2 | (f) |
| 2 | 4 | 1 | (d) |
| | | 2 | (e) |
| | | 3 | (f) |
| | | 4 | (g) |
| 3 | 3 | 1 | (a) |
| | | 2 | (b) |
| | | 3 | (h) |
| 4 | 2 | 1 | (a) |
| | | 2 | (f) |
| 5 | 4 | 1 | (b) |
| | | 2 | (c) |
| | | 3 | (d) |
| | | 4 | (e) |
| 6 | 2 | 1 | (h) |
| | | 2 | (f) |

Due to the intractability of the problem, Xpress solver could not provide an optimal solution after 3600 seconds for any weight combination of any of the instances. Problems 1, 2, 3, and 4 of instance PP1 presented a gap of 79.51%, 79.58%, 73.33% and 72.99% respectively. Feasible solutions (Table 7.27) of problems 1, 2, 3 and 4 and their respective Gantt charts are shown in Figure 7.86, Figure 7.87, Figure 7.88 and Figure 7.89 respectively.

*Table 7.27 PP1 Experimental results for FJSP-2F.*

| Sol | w1 | w2 | w3 | $C_{max}$ | $G$ | $\sum_{j=1}^{n} W_j T_j$ | Gap % | Time (s) |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.3 | 0.3 | 0.4 | 125 | 490 | 3472 | 79.51 | 3600 |
| 2 | 0.7 | 0.1 | 0.2 | 141 | 512 | 3641 | 79.98 | 3600 |
| 3 | 0.1 | 0.6 | 0.3 | 168 | 441 | 3572 | 73.33 | 3600 |
| 4 | 0.3 | 0.5 | 0.2 | 174 | 441 | 3512 | 72.99 | 3600 |

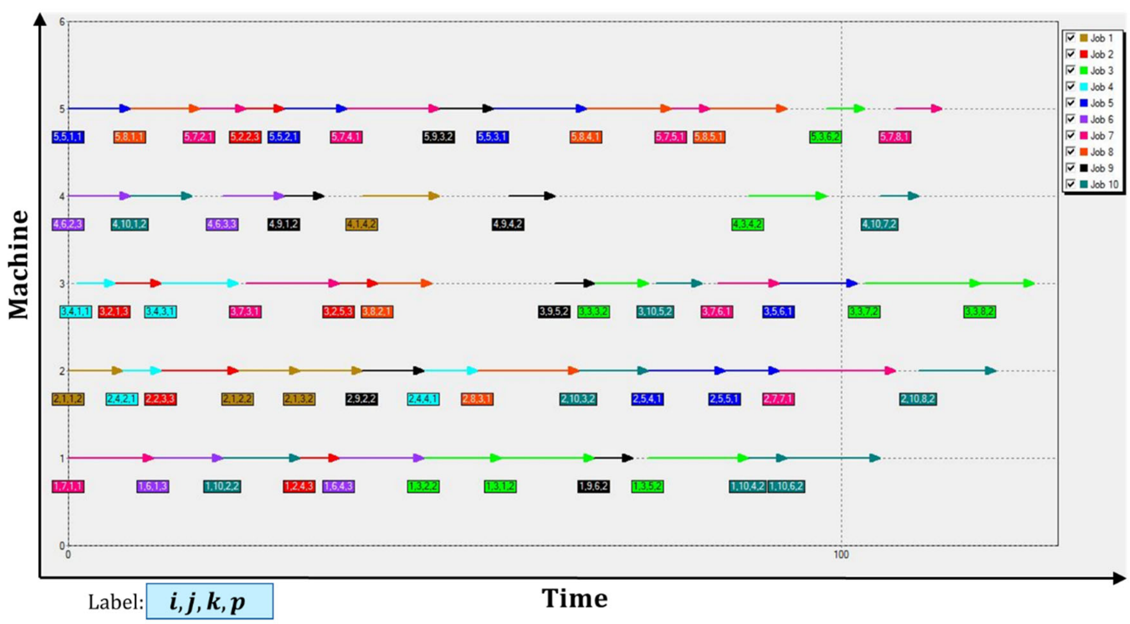



*Figure 7.86 Gantt chart of solution 1.*

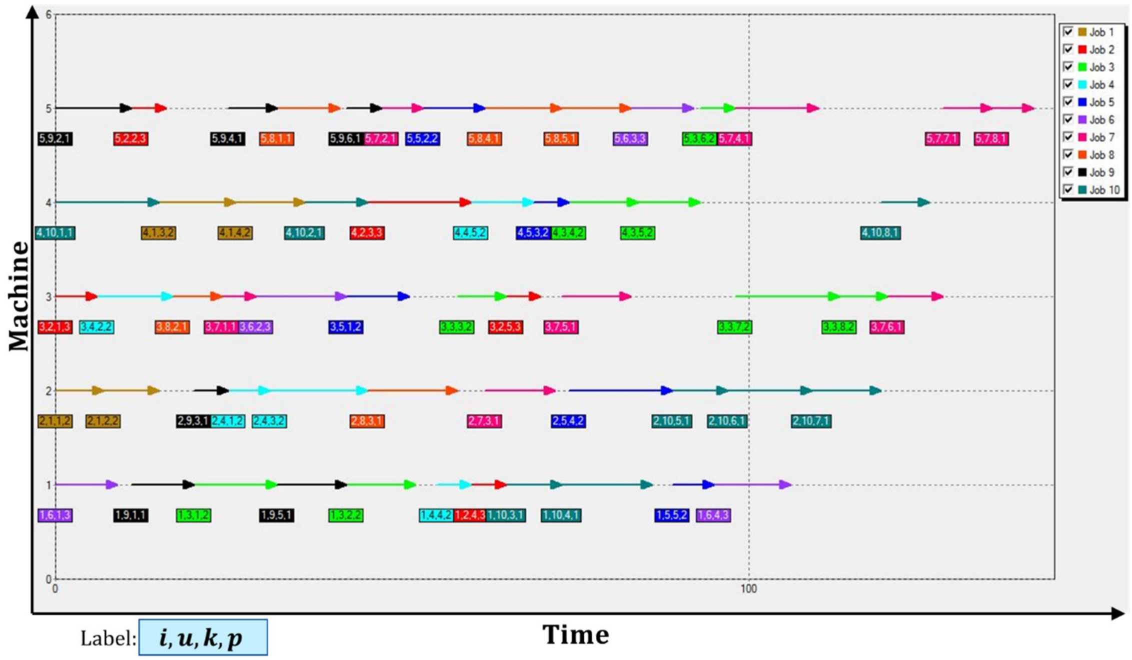



*Figure 7.87 Gantt chart of solution 2.*

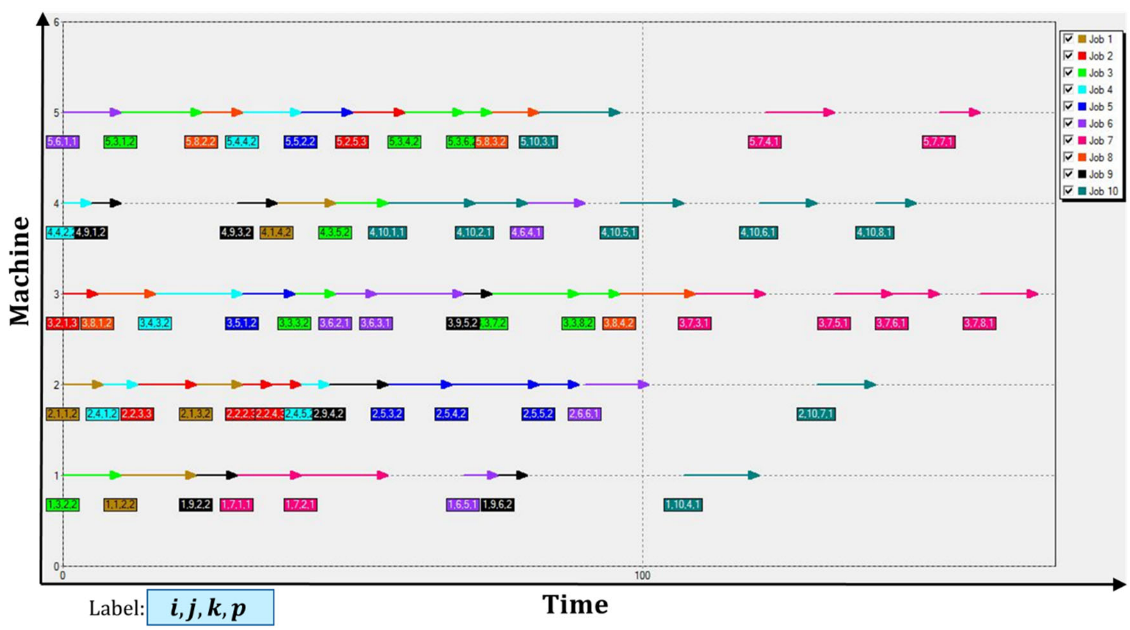



*Figure 7.88 Gantt chart of solution 3.*

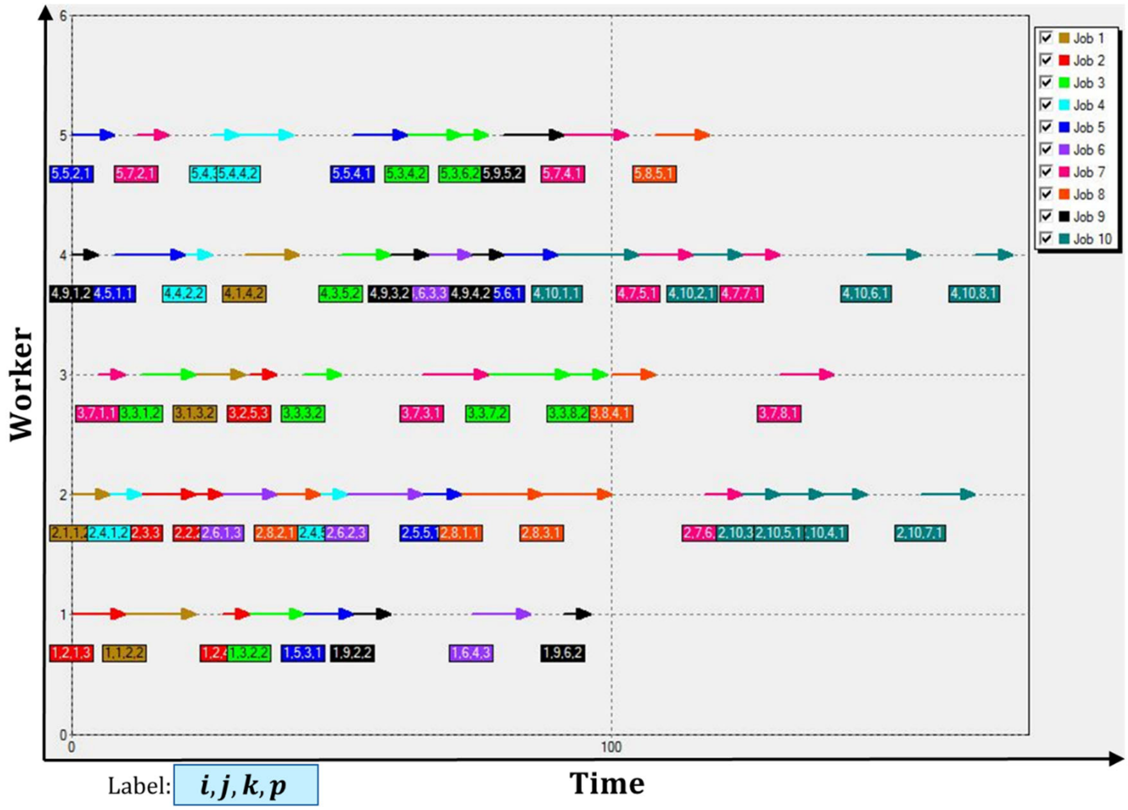


*Figure 7.89 Gantt chart of solution 4.*

In the same manner, problems 1, 2, 3, and 4 of instance PP2 presented a gap of 60.5%, 70.13%, 45.43% and 38.43% respectively. The best integer solutions of problems 1, 2, 3 and 4 are reported in Table 7.28. Similarly, the results found for instance PP3 are shown in Table 7.29.

*Table 7.28 PP2 Experimental results for FJSP-2F.*

| Sol | w1 | w2 | w3 | $C_{max}$ | $G$ | $\sum_{j=1}^{n} W_j T_j$ | Gap % | Time (s) |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.3 | 0.3 | 0.4 | 26 | 219 | 303 | 60.5 | 3600 |
| 2 | 0.7 | 0.1 | 0.2 | 27 | 228 | 311 | 70.13 | 3600 |
| 3 | 0.1 | 0.6 | 0.3 | 31 | 201 | 366 | 45.43 | 3600 |
| 4 | 0.3 | 0.5 | 0.2 | 28 | 201 | 324 | 38.43 | 3600 |

*Table 7.29 PP3 Experimental results for FJSP-2F.*

| Sol | w1 | w2 | w3 | $C_{max}$ | $G$ | $\sum_{j=1}^{n} W_j T_j$ | Gap % | Time (s) |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.3 | 0.3 | 0.4 | 54 | 506 | 805 | 70.59 | 3600 |
| 2 | 0.7 | 0.1 | 0.2 | 56 | 504 | 900 | 78.73 | 3600 |
| 3 | 0.1 | 0.6 | 0.3 | 54 | 471 | 900 | 49.74 | 3600 |
| 4 | 0.3 | 0.5 | 0.2 | 56 | 480 | 823 | 44.58 | 3600 |

# CHAPTER 8. CONCLUSION

## 8.1 Research significance

Manufacturing companies are continually updating their resources by acquiring state-of-the-art technologies. For instance, simultaneous and reconfigurable manufacturing systems have been adopted to respond faster to the unpredictable changes in the demand. The solution methods for the FJSP neglect or assume scenarios that do not represent a real-world situation. For instance, the general FJSP does not take in to account the possibility to select an alternative process plan for a job that better fits the requirements of the shop (resource availability or processing cost). In the same manner, the skilled worker availability is usually neglected.

Additionally, for some jobs, precedence between the operations is given by a directed acyclic graph instead of the sequential manner of the general FJSP. The proposed research aims at incorporating common situations in the manufacturing companies that have not been investigated as worker assignment, sequencing and process plan flexibility, and processing costs. The integration of these elements will contribute to both the literature and the industry with the aim of achieving better resource management.

## 8.2 Limitations

The mathematical model for the FJSPS does not consider setup times and transportation times. However, there are cases where these times cannot be neglected. This scenario represents and opportunity to integrate these elements.

The multi-objective formulation for the DRFJSPS transportation times of the workers are ignored. Additionally, the formulation does not consider worker breaks or shifts.

The last mathematical formulation for the FJSP-2F does not deal with the process plan generation. The process plans and the cost to process and operation in a certain machine is given in advance.

## 8.3 Novelties and contributions

The novelties and contributions summary of this dissertation are presented in Figure 8.1.

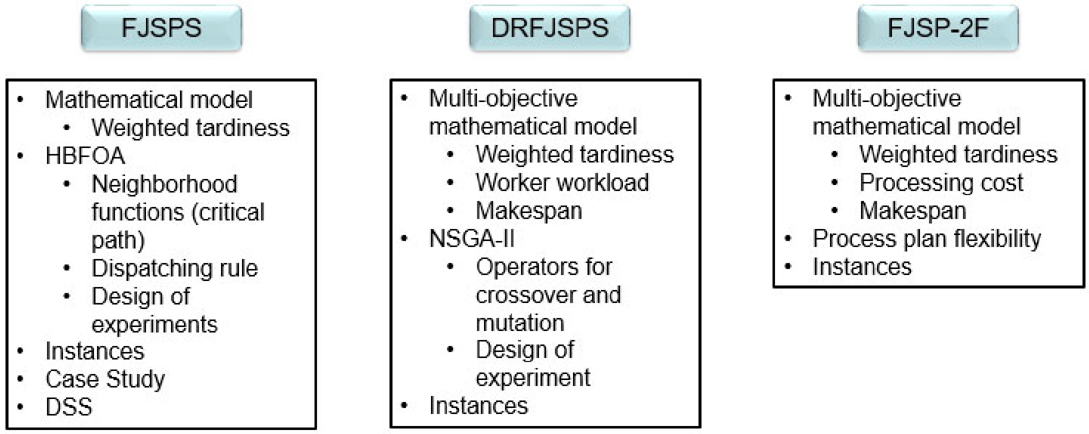



*Figure 8.1 Novelties and contributions.*

The contributions of the proposed work are listed as follow:

- A new mathematical model for the FJSP with flexible sequencing has been devised to minimize weighted tardiness. A different objective to minimize makespan is also considered.
- A HBFOA has been developed to tackle the FJSP with flexible sequencing. The modification to suit the FJSP is done by hybridized SA under the chemotaxis process.
- A new dispatching rule has been devised called minimum number of operations.
- A modified swarming process has been proposed to attract bacteria to more promising solution areas.
- A local search procedure has been proposed using five different neighbourhood function based on the critical path theory.
- A design of experiment has been conducted to obtain the best parameter combination for the HBFOA
- A case study has been introduced where new instances for the FSJP with flexible sequencing have been provided.
- A DSS has been developed on the top of MS Excel using VBA.
- A novel mathematical model for the FJSP with worker selection and flexible sequencing (DRFJSPS) has been developed. The multi-objective formulation minimizes makespan, maximal worker workload and weighted tardiness.

- The implementation of the NSGA-II to efficiently solve the DSFJSPS.
- Three crossover and two mutation operators have been developed.
- New instances for the DRFSJPS with flexible sequencing have been provided.
- A design of experiment has been conducted to fine-tune the NSGA-II parameters.
- A novel mathematical model for the FJSP with sequencing and process plans flexibility (FJSP-2F) has been developed. The multi-objective formulation minimizes makespan, total processing cost and weighted tardiness.
- A new instance that integrates process plan and sequencing flexibility with processing cost has been designed.

## 8.4 Future work

The findings of this research give us the opportunity to continue the expansion of the FJSP. Future research avenues that can be considered are as following:

- Development of a multi-objective metaheuristic for the FJSP with the integration of process plan flexibility (FJSP-2F) needs to be addressed.
- Sequence-dependent setup time can be included in the different problems analyzed in this dissertation.
- Setup cost can be integrated in the study.
- Consideration of worker breaks and shifts needs to be studied.
- Carbon footprint reduction has to be included in the study.
- Implementation of more instances that reflects the features of the (FJSP-2F) has to be considered.
- Integration of a local search method to the NSGA-II.

## 8.5 Conclusion

The current productions environments and the modern manufacturing strategies have required to integrate new features into classical problems that have been neglected over time. To overcome this situation, the integration of sequencing flexibility into the FJSP has been studied in this research.

In this work, three mixed integer programming (MILP) formulations are presented. The first one (FJSPS) is a single objective formulation to minimize the weighted tardiness for the FJSP with sequencing flexibility. A different objective to minimize makespan is also considered. The second formulation (DRFJSPS) covers both machine assignment and worker selection. A multi-objective mathematical model that minimizes makespan, maximal worker workload and weighted tardiness is developed. The third model (FJSP-2F) includes sequencing and process plans flexibility. A multi-objective mathematical model that minimizes makespan, processing cost and weighted tardiness is developed.

Due to the intractable nature of the FJSP with sequencing flexibility, a novel HBFOA has been devised. The algorithm is inspired in the behaviour of E-coli bacterium to seek for food. The HBFOA is further improved by using the SA algorithm under the chemotaxis process to explore more promising solution areas. Furthermore, the HBFOA is enhanced with a local search procedure. The local search is comprised of five neighbourhood functions based on the critical path: three neighbourhood functions for the routing problem and two for the sequencing problem. For the case of the minimization of the weighted tardiness, a case study is analyzed with three instances with sequencing flexibility. A comparison between the HBFOA and the most relevant metaheuristics available in the literature is presented in the case of the minimization of makespan. A design of experiment was executed to fine tune the metaheuristic parameters. The best parameter configuration was reported in this research.

The HBFOA outperformed all classical dispatching rules and the MILP integer solution for all the instances when then weighted tardiness is minimized. For the case of the makespan, the HBFOA outperformed the solutions provided by PVNS and offered comparable results to best solutions of SSPR and HA.

Due to the NP-hardness of the DRFSP, an adaptation of the elitist non-dominated sorting genetic algorithm (NSGA-II) is introduced to minimize conflicting objectives as makespan, total processing costs and weighted tardiness. Moreover, a design of experiments was performed to calibrate the metaheuristics parameters. New instances where developed to implement practical examples in the mathematical model and adapted metaheuristic. The results from the proposed metaheuristic were compared with the solutions of the multi random search algorithm. The comparison demonstrated that the NSGA-II outperformed the solutions provided by the MRSLS.

The proposed research work contributed to the FJSP with the inclusion of elements that have been neglected or assumed. Aspects as sequencing flexibility, process plan flexibility, dual-resource constrained approach, and process plan costs have been integrated to represent a more real manufacturing scenario that will benefit companies and will extend the traditional methods developed to solve the FJSP.

# REFERENCES


Abdi Khalife, M., B. Abbasi, and A. Kamali Dolat abadi. 2010. "A Simulated Annealing Algorithm for Multi Objective Flexible Job Shop Scheduling with Overlapping in Operations." *Journal of Optimization in Industrial Engineering* Volume 3 (Issue 5):17-28.

Ahmadi, E., M. Zandieh, M. Farrokh, and S. M. Emami. 2016. "A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms." *Computers & Operations Research* 73:56-66.

Alvarez-Valdés, R., A. Fuertes, J. M. Tamarit, G. Giménez, and R. Ramos. 2005. "A heuristic to schedule flexible job-shop in a glass factory." *European Journal of Operational Research* 165 (2):525-34.

Amiri, M., M. Zandieh, M. Yazdani, and A. Bagheri. 2010. "A variable neighbourhood search algorithm for the flexible job-shop scheduling problem." *International Journal of Production Research* 48 (19):5671-89.

Birgin, E. G., P. Feofiloff, C. G. Fernandes, E. L. De Melo, M. T. Oshiro, and D. P. Ronconi. 2014. "A MILP model for an extended version of the Flexible Job Shop Problem." *Optimization Letters* 8 (4):1417-31.

Birgin, E. G., J. Ferreira, and D. P. Ronconi. 2015. "List scheduling and beam search methods for the flexible job shop scheduling problem with sequencing flexibility." *European Journal of Operational Research* 247 (2):421-40.

Bowman, E. H. 1959. "The Schedule-Sequencing Problem." *Operations Research* 7 (5):621-4.

Brandimarte, P. 1993. "Routing and scheduling in a flexible job shop by tabu search." *Annals of Operations Research* 41 (3):157-83.

Brandimarte, P., and M. Calderini. 1995. "A hierarchical bicriterion approach to integrated process plan selection and job shop scheduling." *THE INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH* 33 (1):161-81.

Brucker, P., and R. Schlie. 1990. "Job-shop scheduling with multi-purpose machines." *Computing* 45 (4):369-75. doi: 10.1007/BF02238804.

Burke, E. K., and G. Kendall. 2005. *Search methodologies*: Springer.

Chaudhry, I. A., and A. A. Khan. 2016. "A research survey: review of flexible job shop scheduling techniques." *International Transactions in Operational Research* 23 (3):551-91.

Coello, C. A. C., G. B. Lamont, and D. A. Van Veldhuizen. 2007. *Evolutionary algorithms for solving multi-objective problems*. Vol. 5: Springer.

Cormen, T. H., C. E. Leiserson, R. L. Rivest, and C. Stein. 2001. *Introduction to algorithms*. Vol. 6: MIT press Cambridge.

De Giovanni, L., and F. Pezzella. 2010. "An improved genetic algorithm for the distributed and flexible job-shop scheduling problem." *European Journal of Operational Research* 200 (2):395-408.

Deb, K., S. Agrawal, A. Pratap, and T. Meyarivan. 2000. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. Paper presented at the International Conference on Parallel Problem Solving From Nature.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. 2002. "A fast and elitist multiobjective genetic algorithm: NSGA-II." *IEEE transactions on evolutionary computation* 6 (2):182-97.

Demir, Y., and S. K. İşleyen. 2014. "An effective genetic algorithm for flexible job-shop scheduling with overlapping in operations." *International Journal of Production Research* 52 (13):3905-21. doi: 10.1080/00207543.2014.889328.

Demir, Y., and S. Kürşat İşleyen. 2013. "Evaluation of mathematical models for flexible job-shop scheduling problems." *Applied Mathematical Modelling* 37 (3):977-88. doi: http://dx.doi.org/10.1016/j.apm.2012.03.020.

Driss, I., K. N. Mouss, and A. Laggoun. 2015. "A new genetic algorithm for flexible job-shop scheduling problems." *Journal of Mechanical Science and Technology* 29 (3):1273-81. doi: 10.1007/s12206-015-0242-7.

ElMaraghy, H. A. 2005. "Flexible and reconfigurable manufacturing systems paradigms." *International journal of flexible manufacturing systems* 17 (4):261-76.

Fattahi, P., F. Jolai, and J. Arkat. 2009. "Flexible job shop scheduling with overlapping in operations." *Applied Mathematical Modelling* 33 (7):3076-87.

Fattahi, P., M. S. Mehrabad, and F. Jolai. 2007. "Mathematical modeling and heuristic approaches to flexible job shop scheduling problems." *Journal of Intelligent Manufacturing* 18 (3):331-42.

Gan, P. Y., and K. Lee. 2002. "Scheduling of flexible-sequenced process plans in a mould manufacturing shop." *The International Journal of Advanced Manufacturing Technology* 20 (3):214-22.

Gao, J., M. Gen, and L. Sun. 2006. "Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm." *Journal of Intelligent Manufacturing* 17 (4):493-507.

Gao, K. Z., P. N. Suganthan, T. J. Chua, C. S. Chong, T. X. Cai, and Q. K. Pan. 2015. "A two-stage artificial bee colony algorithm scheduling flexible job-shop scheduling problem with new job insertion." *Expert Systems with Applications* 42 (21):7652-63. doi: http://dx.doi.org/10.1016/j.eswa.2015.06.004.

Gao, L., C. Peng, C. Zhou, and P. Li. 2006. Solving flexible job shop scheduling problem using general particle swarm optimization. Paper presented at the Proceedings of the 36th CIE conference on computers & industrial engineering.

Garey, M. R., D. S. Johnson, and R. Sethi. 1976. "THE COMPLEXITY OF FLOWSHOP AND JOBSHOP SCHEDULING." *Mathematics of Operations Research* 1 (2):117-29.

Gen, M., J. Gao, and L. Lin. 2009. "Multistage-based genetic algorithm for flexible job-shop scheduling problem." In *Intelligent and evolutionary systems*, 183-96. Springer.

Gomes*, M., A. Barbosa-Povoa, and A. Novais. 2005. "Optimal scheduling for flexible job shop operation." *International Journal of Production Research* 43 (11):2323-53.

Gong, G., Q. Deng, X. Gong, W. Liu, and Q. Ren. 2018. "A new double flexible job-shop scheduling problem integrating processing time, green production, and human factor indicators." *Journal of Cleaner Production* 174:560-76.

González, Miguel A., Camino R. Vela, and R. Varela. 2015. "Scatter search with path relinking for the flexible job shop scheduling problem." *European Journal of Operational Research* 245 (1):35-45. doi: http://dx.doi.org/10.1016/j.ejor.2015.02.052.

Hernández-Ocana, B., M. D. P. Pozos-Parra, and E. Mezura-Montes. 2016. "Improved modified bacterial foraging optimization algorithm to solve constrained numerical optimization problems." *Applied Mathematics and Information Sciences* 10 (2):607-22.

Ho, N. B., J. C. Tay, and E. M. K. Lai. 2007. "An effective architecture for learning and evolving flexible job-shop schedules." *European Journal of Operational Research* 179 (2):316-33. doi: http://dx.doi.org/10.1016/j.ejor.2006.04.007.

Huang, R.-H., C.-L. Yang, and W.-C. Cheng. 2013. "Flexible job shop scheduling with due window—a two-pheromone ant colony approach." *International Journal of Production Economics* 141 (2):685-97. doi: http://dx.doi.org/10.1016/j.ijpe.2012.10.011.

Hurink, J., B. Jurisch, and M. Thole. 1994. "Tabu search for the job-shop scheduling problem with multi-purpose machines." *Operations-Research-Spektrum* 15 (4):205-15.

Imanipour, N. 2006. Modeling & solving flexible job shop problem with sequence dependent setup times. Paper presented at the Service Systems and Service Management, 2006 International Conference on.

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

———. 2002b. "Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic." *Mathematics and computers in simulation* 60 (3):245-76.

Kalpakjian, S. 2001. *Manufacturing engineering and technology*: Pearson Education India.

Kaplanoğlu, V. 2016. "An object-oriented approach for multi-objective flexible job-shop scheduling problem." *Expert Systems with Applications* 45:71-84. doi: http://dx.doi.org/10.1016/j.eswa.2015.09.050.

Kim, K.-H., and P. Egbelu. 1999. "Scheduling in a production environment with multiple process plans per job." *International Journal of Production Research* 37 (12):2725-53.

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi. 1983. "Optimization by Simulated Annealing." *Science* 220 (4598):671.

Lee, Y. H., C. S. Jeong, and C. Moon. 2002. "Advanced planning and scheduling with outsourcing in manufacturing supply chain." *Computers & Industrial Engineering* 43 (1–2):351-74. doi: http://dx.doi.org/10.1016/S0360-8352(02)00079-7.

Li, X., and L. Gao. 2016. "An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem." *International Journal of Production Economics* 174:93-110.

Lopez, P., and F. Roubellat. 2013. *Production scheduling*: John Wiley & Sons.

Loukil, T., J. Teghem, and P. Fortemps. 2007. "A multi-objective production scheduling case study solved by simulated annealing." *European Journal of Operational Research* 179 (3):709-22.

Low, C., and T.-H. Wu. 2001. "Mathematical modelling and heuristic approaches to operation scheduling problems in an FMS environment." *International Journal of Production Research* 39 (4):689-708. doi: 10.1080/00207540150504403.

Manne, A. S. 1960. "On the Job-Shop Scheduling Problem." *Operations Research* 8 (2):219-23.

Mastrolilli, M., and L. M. Gambardella. 2000. "Effective neighborhood functions for the flexible job shop problem: Appendix."

Montgomery, D. C. 2008. *Design and analysis of experiments*: John Wiley & Sons.

Nahmias, S. 2009. *Production and operations analysis*. 6th ed.. ed. New York, NY: New York, NY : McGraw-Hill/Irwin.

Özgüven, C., L. Özbakır, and Y. Yavuz. 2010. "Mathematical models for job-shop scheduling problems with routing and process plan flexibility." *Applied Mathematical Modelling* 34 (6):1539-48. doi: http://dx.doi.org/10.1016/j.apm.2009.09.002.

Passino, K. M. 2002. "Biomimicry of bacterial foraging for distributed optimization and control." *IEEE Control Systems* 22 (3):52-67. doi: 10.1109/MCS.2002.1004010.

Pezzella, F., G. Morganti, and G. Ciaschetti. 2008. "A genetic algorithm for the flexible job-shop scheduling problem." *Computers & Operations Research* 35 (10):3202-12.

Pinedo, M. 2005. *Planning and scheduling in manufacturing and services*: Springer.

Roshanaei, V., A. Azab, and H. ElMaraghy. 2013. "Mathematical modelling and a meta-heuristic for flexible job shop scheduling." *International Journal of Production Research* 51 (20):6247-74. doi: 10.1080/00207543.2013.827806.

Rossi, A., and G. Dini. 2007. "Flexible job-shop scheduling with routing flexibility and separable setup times using ant colony optimisation method." *Robotics and Computer-Integrated Manufacturing* 23 (5):503-16. doi: http://dx.doi.org/10.1016/j.rcim.2006.06.004.

Shahsavari-Pour, N., and B. Ghasemishabankareh. 2013. "A novel hybrid meta-heuristic algorithm for solving multi objective flexible job shop scheduling." *Journal of Manufacturing Systems* 32 (4):771-80. doi: http://dx.doi.org/10.1016/j.jmsy.2013.04.015.

Shen, W., L. Wang, and Q. Hao. 2006. "Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 36 (4):563-77.

Srinivas, N., and K. Deb. 1994. "Muiltiobjective optimization using nondominated sorting in genetic algorithms." *Evolutionary computation* 2 (3):221-48.

Thomalla, C. S. 2001. "Job shop scheduling with alternative process plans." *International Journal of Production Economics* 74 (1):125-34.

Unlu, Y., and S. J. Mason. 2010. "Evaluation of mixed integer programming formulations for non-preemptive parallel machine scheduling problems." *Computers & Industrial Engineering* 58 (4):785-800. doi: https://doi.org/10.1016/j.cie.2010.02.012.

Van Laarhoven, P. J., E. H. Aarts, and J. K. Lenstra. 1992. "Job shop scheduling by simulated annealing." *Operations Research* 40 (1):113-25.

Vilcot, G., and J.-C. Billaut. 2008. "A tabu search and a genetic algorithm for solving a bicriteria general job shop scheduling problem." *European Journal of Operational Research* 190 (2):398-411.

Vital Soto, A., N. T. Chowdhury, M. Z. Allahyari, A. Azab, and M. F. Baki. 2017. "Mathematical modeling and hybridized evolutionary LP local search method for lot-sizing with supplier selection, inventory shortage, and quantity discounts." *Computers & Industrial Engineering* 109:96-112. doi: https://doi.org/10.1016/j.cie.2017.04.027.

Wagner, H. M. 1959. "An integer linear-programming model for machine scheduling." *Naval Research Logistics Quarterly* 6 (2):131-40.

Wang, L., G. Zhou, Y. Xu, S. Wang, and M. Liu. 2012. "An effective artificial bee colony algorithm for the flexible job-shop scheduling problem." *The International Journal of Advanced Manufacturing Technology* 60 (1):303-15. doi: 10.1007/s00170-011-3610-1.

Wu, C., N. Zhang, J. Jiang, J. Yang, and Y. Liang. 2007. "Improved Bacterial Foraging Algorithms and Their Applications to Job Shop Scheduling Problems." In *Adaptive and Natural Computing Algorithms: 8th International Conference, ICANNGA 2007, Warsaw, Poland, April 11-14, 2007, Proceedings, Part I*, edited by Bartlomiej Beliczynski, Andrzej Dzielinski, Marcin Iwanowski and Bernardete Ribeiro, 562-9. Berlin, Heidelberg: Springer Berlin Heidelberg.

Xia, W., and Z. Wu. 2005. "An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems." *Computers & Industrial Engineering* 48 (2):409-25.

Xing, L.-N., Y.-W. Chen, P. Wang, Q.-S. Zhao, and J. Xiong. 2010. "A knowledge-based ant colony optimization for flexible job shop scheduling problems." *Applied Soft Computing* 10 (3):888-96.

Yazdani, M., M. Amiri, and M. Zandieh. 2010. "Flexible job-shop scheduling with parallel variable neighborhood search algorithm." *Expert Systems with Applications* 37 (1):678-87.

Zhang, Q., H. Manier, and M. A. Manier. 2012. "A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times." *Computers & Operations Research* 39 (7):1713-23. doi: http://dx.doi.org/10.1016/j.cor.2011.10.007.

Zhao, F., X. Jiang, C. Zhang, and J. Wang. 2015. "A chemotaxis-enhanced bacterial foraging algorithm and its application in job shop scheduling problem." *International Journal of*

*Computer Integrated Manufacturing* 28 (10):1106-21. doi: 10.1080/0951192X.2014.961962.

Zheng, X.-l., and L. Wang. 2016. "A knowledge-guided fruit fly optimization algorithm for dual resource constrained flexible job-shop scheduling problem." *International Journal of Production Research*:1-13. doi: 10.1080/00207543.2016.1170226.

# APPENDIX A. INSTANCES

*Table A.1 Case I 10 x 15 with 63 operations.*

| Jobs $j$ | Op. $o_{jk}$ | $P_{1jk}$ | $P_{2jk}$ | $P_{3jk}$ | $P_{4jk}$ | $P_{5jk}$ | $P_{6jk}$ | $P_{7jk}$ | $P_{8jk}$ | $P_{9jk}$ | $P_{10jk}$ | $P_{11jk}$ | $P_{12jk}$ | $P_{13jk}$ | $P_{14jk}$ | $P_{15jk}$ | Due Date $d_j$ | Weight $W_j$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | $o_{1,1}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 19 | 2 |
| | $o_{1,2}$ | 0 | 0 | 4 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{1,3}$ | 0 | 0 | 0 | 0 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{1,4}$ | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 5 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{1,5}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 4 | 0 | 0 | 0 | | |
| | $o_{1,6}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{1,7}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 5 | 0 | | |
| | $o_{1,8}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 2 | $o_{2,1}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 | 3 |
| | $o_{2,2}$ | 0 | 0 | 4 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{2,3}$ | 0 | 0 | 0 | 0 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{2,4}$ | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 6 | 4 | 4 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{2,5}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 4 | 0 | 0 | 0 | | |
| | $o_{2,6}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{2,7}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 5 | 0 | | |
| | $o_{2,8}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 3 | $o_{3,1}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 23 | 1 |
| | $o_{3,2}$ | 0 | 0 | 4 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{3,3}$ | 0 | 0 | 0 | 0 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{3,4}$ | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 6 | 4 | 4 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{3,5}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 4 | 0 | 0 | 0 | | |
| | $o_{3,6}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{3,7}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 5 | 0 | | |
| | $o_{3,8}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 4 | $o_{4,1}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 19 | 1 |
| | $o_{4,2}$ | 0 | 0 | 4 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{4,3}$ | 0 | 0 | 0 | 0 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{4,4}$ | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 6 | 4 | 4 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{4,5}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 4 | 0 | 0 | 0 | | |
| | $o_{4,6}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 5 | $o_{5,1}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 4 |
| | $o_{5,2}$ | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{5,3}$ | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{5,4}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 6 | $o_{6,1}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 | 1 |
| | $o_{6,2}$ | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{6,3}$ | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{6,4}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{6,5}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 5 | 0 | | |
| | $o_{6,6}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 7 | $o_{7,1}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 1 |
| | $o_{7,2}$ | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{7,3}$ | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{7,4}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |

*Table A.2 Case I 10 x 15 with 63 operations (continued).*

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $o_{7,5}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 5 | 0 | | |
| | $o_{7,6}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 8 | $o_{8,1}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 17 | 1 |
| | $o_{8,2}$ | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{8,3}$ | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{8,4}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{8,5}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 5 | 0 | | |
| | $o_{8,6}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 9 | $o_{9,1}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 19 | 3 |
| | $o_{9,2}$ | 0 | 0 | 4 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{9,3}$ | 0 | 0 | 0 | 0 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{9,4}$ | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 5 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{9,5}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 10 | $o_{10,1}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 | 2 |
| | $o_{10,2}$ | 0 | 0 | 4 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{10,3}$ | 0 | 0 | 0 | 0 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{10,4}$ | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 5 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{10,5}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{10,6}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | | |

*Table A.3 Case II 5 x 12 with 25 operations.*

| Jobs $j$ | Op. $o_{jk}$ | Machines $i$ | | | | | | | | | | | | Due Date $d_j$ | Weight $W_j$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $P_{1jk}$ | $P_{2jk}$ | $P_{3jk}$ | $P_{4jk}$ | $P_{5jk}$ | $P_{6jk}$ | $P_{7jk}$ | $P_{8jk}$ | $P_{9jk}$ | $P_{10jk}$ | $P_{11jk}$ | $P_{12jk}$ | | |
| 1 | $o_{1,1}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 6 |
| | $o_{1,2}$ | 0 | 0 | 4 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{1,3}$ | 0 | 0 | 0 | 0 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{1,4}$ | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 6 | 4 | 4 | 0 | 0 | | |
| | $o_{1,5}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 4 | | |
| | $o_{1,6}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 2 | $o_{2,1}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 3 |
| | $o_{2,2}$ | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{2,3}$ | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 2 | 2 | 0 | 0 | | |
| | $o_{2,4}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 3 | $o_{3,1}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 19 | 3 |
| | $o_{3,2}$ | 0 | 0 | 4 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{3,3}$ | 0 | 0 | 0 | 0 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{3,4}$ | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 5 | 3 | 3 | 0 | 0 | | |
| | $o_{3,5}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 4 | $o_{4,1}$ | 6 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 27 | 5 |
| | $o_{4,2}$ | 0 | 0 | 4 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{4,3}$ | 0 | 0 | 0 | 0 | 4 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{4,4}$ | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 7 | 5 | 5 | 0 | 0 | | |
| | $o_{4,5}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 5 | | |
| | $o_{4,6}$ | 6 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 5 | $o_{5,1}$ | 7 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 17 | 1 |
| | $o_{5,2}$ | 0 | 0 | 0 | 0 | 4 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{5,3}$ | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 5 | 4 | 4 | 0 | 0 | | |
| | $o_{5,4}$ | 7 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |

*Table A.4 Case III 8 x 14 with 44 operations.*

| Jobs $j$ | Op. $o_{jk}$ | Machines $i$ | | | | | | | | | | | | | Due Date | Weight |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $P_{1jk}$ | $P_{2jk}$ | $P_{3jk}$ | $P_{4jk}$ | $P_{5jk}$ | $P_{6jk}$ | $P_{7jk}$ | $P_{8jk}$ | $P_{9jk}$ | $P_{10jk}$ | $P_{11jk}$ | $P_{12jk}$ | $P_{13jk}$ | $d_j$ | $W_j$ |
| 1 | $o_{1,1}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 19 | 2 |
| | $o_{1,2}$ | 0 | 0 | 4 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{1,3}$ | 0 | 0 | 0 | 0 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{1,4}$ | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 6 | 4 | 4 | 0 | 0 | 0 | | |
| | $o_{1,5}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 4 | 0 | | |
| | $o_{1,6}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 2 | $o_{2,1}$ | 7 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 35 | 2 |
| | $o_{2,2}$ | 0 | 0 | 6 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{2,3}$ | 0 | 0 | 0 | 0 | 5 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{2,4}$ | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 8 | 6 | 6 | 0 | 0 | 0 | | |
| | $o_{2,5}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 6 | 0 | | |
| | $o_{2,6}$ | 7 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 3 | $o_{3,1}$ | 4 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 12 | 4 |
| | $o_{3,2}$ | 0 | 0 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{3,3}$ | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{3,4}$ | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 5 | 3 | 3 | 0 | 0 | 0 | | |
| | $o_{3,5}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 0 | | |
| | $o_{3,6}$ | 4 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 4 | $o_{4,1}$ | 6 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 28 | 5 |
| | $o_{4,2}$ | 0 | 0 | 5 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{4,3}$ | 0 | 0 | 0 | 0 | 4 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{4,4}$ | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 7 | 5 | 5 | 0 | 0 | 0 | | |
| | $o_{4,5}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 5 | 0 | | |
| | $o_{4,6}$ | 6 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 5 | $o_{5,1}$ | 6 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 | 1 |
| | $o_{5,2}$ | 0 | 0 | 0 | 0 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{5,3}$ | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 3 | 3 | 0 | 0 | 0 | | |
| | $o_{5,4}$ | 6 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 6 | $o_{6,1}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 5 |
| | $o_{6,2}$ | 0 | 0 | 4 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{6,3}$ | 0 | 0 | 0 | 0 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{6,4}$ | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 5 | 3 | 3 | 0 | 0 | 0 | | |
| | $o_{6,5}$ | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 7 | $o_{7,1}$ | 6 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 22 | 4 |
| | $o_{7,2}$ | 0 | 0 | 5 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{7,3}$ | 0 | 0 | 0 | 0 | 4 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{7,4}$ | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 6 | 4 | 4 | 0 | 0 | 0 | | |
| | $o_{7,5}$ | 6 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 8 | $o_{8,1}$ | 7 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 31 | 4 |
| | $o_{8,2}$ | 0 | 0 | 5 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{8,3}$ | 0 | 0 | 0 | 0 | 4 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{8,4}$ | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 6 | 4 | 4 | 0 | 0 | 0 | | |
| | $o_{8,5}$ | 7 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | $o_{8,6}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | | |

*Table A.5 DR1 10 x 15 x 10 with 63 operations.*

| Jobs | | Machine 1 | | | | | | | | | | Machine 2 | | | | | | | | | | Machine 3 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $j$ | 85.57 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 |
| | 1,1 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - |
| | 1,2 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - |
| | 1,3 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 |
| 1 | 1,4 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 |
| | 1,5 | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - |
| | 1,6 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - |
| | 1,7 | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - |
| | 1,8 | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 |
| | 2,1 | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 |
| | 2,2 | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - |
| | 2,3 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 |
| | 2,4 | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 |
| 2 | 2,5 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 |
| | 2,6 | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 |
| | 2,7 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 |
| | 2,8 | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 |
| | 3,1 | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 |
| | 3,2 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 |
| | 3,3 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 |
| | 3,4 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 |
| 3 | 3,5 | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 |
| | 3,6 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - |
| | 3,7 | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - |
| | 3,8 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - |
| | 4,1 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 |
| | 4,2 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 |
| 4 | 4,3 | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 |
| | 4,4 | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 |
| | 4,5 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 |
| | 4,6 | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 |
| | 4,7 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - |
| 5 | 4,8 | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - |
| | 4,9 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - |
| | 4,10 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - |

*Table A.6 DR1 10 x 15 x 10 with 63 operations (continued).*

| Jobs | | Machine 4 | | | | | | | | | | Machine 5 | | | | | | | | | | Machine 6 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *j* | 85.57 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 |
| 1 | 1,1 | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 |
| | 1,2 | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 |
| | 1,3 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 |
| | 1,4 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 |
| | 1,5 | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - |
| | 1,6 | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 |
| | 1,7 | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 |
| | 1,8 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 |
| 2 | 2,1 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 |
| | 2,2 | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - |
| | 2,3 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 |
| | 2,4 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - |
| | 2,5 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - |
| | 2,6 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - |
| | 2,7 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 |
| | 2,8 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - |
| 3 | 3,1 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 |
| | 3,2 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 |
| | 3,3 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - |
| | 3,4 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - |
| | 3,5 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - |
| | 3,6 | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - |
| | 3,7 | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - |
| | 3,8 | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 |
| 4 | 4,1 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 |
| | 4,2 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 |
| | 4,3 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - |
| | 4,4 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - |
| | 4,5 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 |
| | 4,6 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 |
| 5 | 4,7 | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 |
| | 4,8 | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 |
| | 4,9 | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - |
| | 4,10 | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - |

*Table A.7 DR1 10 x 15 x 10 with 63 operations (continued).*

| Jobs | | | | | | Machine 7 | | | | | | | | Machine 8 | | | | | | | | | | | | Machine 9 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *j* | 85.57 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 |
| | 1,1 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - |
| | 1,2 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - |
| | 1,3 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 |
| 1 | 1,4 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - |
| | 1,5 | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - |
| | 1,6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 |
| | 1,7 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 |
| | 1,8 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - |
| | 2,1 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 |
| | 2,2 | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - |
| | 2,3 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 |
| 2 | 2,4 | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - |
| | 2,5 | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 |
| | 2,6 | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 |
| | 2,7 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 |
| | 2,8 | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 |
| | 3,1 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 |
| | 3,2 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 |
| | 3,3 | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 |
| 3 | 3,4 | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 |
| | 3,5 | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 |
| | 3,6 | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - |
| | 3,7 | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - |
| | 3,8 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 |
| | 4,1 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - |
| | 4,2 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 |
| 4 | 4,3 | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 |
| | 4,4 | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 |
| | 4,5 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 |
| | 4,6 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 |
| | 4,7 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - |
| 5 | 4,8 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 |
| | 4,9 | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 |
| | 4,10 | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 |

*Table A.8 DR1 10 x 15 x 10 with 63 operations (continued).*

| Jobs | | Machine 10 | | | | | | | | | | Machine 11 | | | | | | | | | | Machine 12 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *j* | 85.57 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 |
| 1 | 1,1 | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - |
| | 1,2 | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 |
| | 1,3 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - |
| | 1,4 | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - |
| | 1,5 | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - |
| | 1,6 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 |
| | 1,7 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - |
| | 1,8 | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - |
| 2 | 2,1 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - |
| | 2,2 | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 |
| | 2,3 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - |
| | 2,4 | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 |
| | 2,5 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 |
| | 2,6 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - |
| | 2,7 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 |
| | 2,8 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - |
| 3 | 3,1 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - |
| | 3,2 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 |
| | 3,3 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 |
| | 3,4 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 |
| | 3,5 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - |
| | 3,6 | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 |
| | 3,7 | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 |
| | 3,8 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - |
| 4 | 4,1 | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 |
| | 4,2 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - |
| | 4,3 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - |
| | 4,4 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - |
| | 4,5 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 |
| | 4,6 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - |
| 5 | 4,7 | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 |
| | 4,8 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - |
| | 4,9 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - |
| | 4,10 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - |

*Table A.9 DR1 10 x 15 x 10 with 63 operations (continued).*

| Jobs | | Machine 13 | | | | | | | | | | Machine 14 | | | | | | | | | | Machine 15 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *j* | 85.57 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 |
| 1 | 1,1 | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 |
| | 1,2 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 |
| | 1,3 | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 |
| | 1,4 | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - |
| | 1,5 | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 |
| | 1,6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - |
| | 1,7 | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - |
| | 1,8 | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - |
| 2 | 2,1 | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - |
| | 2,2 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 |
| | 2,3 | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - |
| | 2,4 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 |
| | 2,5 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - |
| | 2,6 | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 |
| | 2,7 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - |
| | 2,8 | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - |
| 3 | 3,1 | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 |
| | 3,2 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 |
| | 3,3 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 |
| | 3,4 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - |
| | 3,5 | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 |
| | 3,6 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - |
| | 3,7 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 |
| | 3,8 | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 |
| 4 | 4,1 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 |
| | 4,2 | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - |
| | 4,3 | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - |
| | 4,4 | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 |
| | 4,5 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - |
| | 4,6 | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 |
| 5 | 4,7 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - |
| | 4,8 | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 |
| | 4,9 | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 |
| | 4,10 | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 |

*Table A.10 DR1 10 x 15 x 10 with 63 operations (continued).*

| Jobs | | Machine 1 | | | | | | | | | | Machine 2 | | | | | | | | | | Machine 3 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *j* | 85.57 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 |
| 6 | 4,11 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - |
| | 4,12 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 |
| | 4,13 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - |
| | 4,14 | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - |
| | 4,15 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 |
| | 4,16 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 |
| 7 | 4,17 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - |
| | 4,18 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 |
| | 4,19 | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - |
| | 4,20 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 |
| | 4,21 | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - |
| | 4,22 | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 |
| 8 | 4,23 | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 |
| | 4,24 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 |
| | 4,25 | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 |
| | 4,26 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 |
| | 4,27 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 |
| | 4,28 | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 |
| 9 | 4,29 | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 |
| | 4,30 | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 |
| | 4,31 | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - |
| | 4,32 | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - |
| | 4,33 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 |
| 10 | 4,34 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - |
| | 4,35 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 |
| | 4,36 | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 |
| | 4,37 | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 |
| | 4,38 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 |
| | 4,39 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 |

*Table A.11 DR1 10 x 15 x 10 with 63 operations (continued).*

| Jobs | | Machine 4 | | | | | | | | | | Machine 5 | | | | | | | | | | Machine 6 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *j* | 85.57 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 |
| 6 | 4,11 | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 |
| | 4,12 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - |
| | 4,13 | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - |
| | 4,14 | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - |
| | 4,15 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 |
| | 4,16 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - |
| 7 | 4,17 | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - |
| | 4,18 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - |
| | 4,19 | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 |
| | 4,20 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - |
| | 4,21 | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 |
| | 4,22 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 |
| 8 | 4,23 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - |
| | 4,24 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 |
| | 4,25 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - |
| | 4,26 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - |
| | 4,27 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 |
| | 4,28 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 |
| 9 | 4,29 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 |
| | 4,30 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - |
| | 4,31 | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 |
| | 4,32 | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 |
| | 4,33 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - |
| 10 | 4,34 | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 |
| | 4,35 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - |
| | 4,36 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - |
| | 4,37 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - |
| | 4,38 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 |
| | 4,39 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - |

*Table A.12 DR1 10 x 15 x 10 with 63 operations (continued).*

| Jobs | | Machine 7 | | | | | | | | | | Machine 8 | | | | | | | | | | Machine 9 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *j* | 85.57 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 |
| 6 | 4,11 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 |
| | 4,12 | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 |
| | 4,13 | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - |
| | 4,14 | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 |
| | 4,15 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - |
| | 4,16 | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - |
| 7 | 4,17 | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - |
| | 4,18 | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - |
| | 4,19 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 |
| | 4,20 | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - |
| | 4,21 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 |
| | 4,22 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - |
| 8 | 4,23 | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 |
| | 4,24 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - |
| | 4,25 | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - |
| | 4,26 | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 |
| | 4,27 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 |
| | 4,28 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 |
| 9 | 4,29 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - |
| | 4,30 | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 |
| | 4,31 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - |
| | 4,32 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 |
| | 4,33 | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 |
| 10 | 4,34 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 |
| | 4,35 | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - |
| | 4,36 | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - |
| | 4,37 | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 |
| | 4,38 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - |
| | 4,39 | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 |

*Table A.13 DR1 10 x 15 x 10 with 63 operations (continued).*

| Jobs | | Machine 10 | | | | | | | | | | Machine 11 | | | | | | | | | | Machine 12 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *j* | 85.57 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 |
| 6 | 4,11 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 |
| | 4,12 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 |
| | 4,13 | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - |
| | 4,14 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - |
| | 4,15 | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - |
| | 4,16 | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 |
| 7 | 4,17 | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 |
| | 4,18 | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - |
| | 4,19 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 |
| | 4,20 | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 |
| | 4,21 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 |
| | 4,22 | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 |
| 8 | 4,23 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 |
| | 4,24 | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 |
| | 4,25 | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 |
| | 4,26 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 |
| | 4,27 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 |
| | 4,28 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 |
| 9 | 4,29 | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 |
| | 4,30 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - |
| | 4,31 | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - |
| | 4,32 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - |
| | 4,33 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 |
| 10 | 4,34 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 |
| | 4,35 | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 |
| | 4,36 | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 |
| | 4,37 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 |
| | 4,38 | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 |
| | 4,39 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - |

*Table A.14 DR1 10 x 15 x 10 with 63 operations (continued).*

| Jobs | | Machine 13 | | | | | | | | | | Machine 14 | | | | | | | | | | Machine 15 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *j* | 85.57 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 | u9 | u10 |
| 6 | 4,11 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 |
| | 4,12 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 |
| | 4,13 | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - |
| | 4,14 | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 |
| | 4,15 | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 |
| | 4,16 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 |
| 7 | 4,17 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 |
| | 4,18 | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - |
| | 4,19 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 |
| | 4,20 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - |
| | 4,21 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - |
| | 4,22 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - |
| 8 | 4,23 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 |
| | 4,24 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - |
| | 4,25 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 |
| | 4,26 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 |
| | 4,27 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - |
| | 4,28 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - | 18 | - | - | - | - | 19 | - | 6 | 7 | - | 13 | - | - |
| 9 | 4,29 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - | - | - | - | 9 | - | 11 | - | - | 10 | 19 | 10 | 13 | - |
| | 4,30 | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - | 15 | 18 | - | - | 18 | - | 20 | - | 13 | - | - | - | - |
| | 4,31 | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 | - | 18 | - | 10 | 7 | 7 | - | - | 9 | 11 | 17 | 19 | - |
| | 4,32 | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 | - | - | - | 5 | 14 | - | - | 8 | - | 13 | - | - | 19 |
| | 4,33 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 | 12 | 7 | 15 | 14 | - | - | - | - | - | 5 | - | 20 | 9 |
| 10 | 4,34 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - | 5 | 18 | - | 6 | 10 | - | 6 | 12 | - | 19 | - | 11 | 5 |
| | 4,35 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - | 11 | 14 | - | 20 | - | 20 | - | - | 6 | 14 | - | 13 | - |
| | 4,36 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 | - | 8 | 12 | - | 5 | - | - | - | 16 | 11 | 15 | 5 | - |
| | 4,37 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 | 12 | 5 | 17 | 5 | 11 | 9 | - | - | 14 | 9 | - | 16 | 7 |
| | 4,38 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 | 16 | 14 | - | 8 | 18 | 5 | 15 | 7 | 17 | 9 | 8 | - | 11 |
| | 4,39 | - | - | 19 | - | 6 | 7 | - | 13 | - | - | 15 | - | 20 | - | 8 | 14 | 5 | - | 9 | 7 | 19 | - | - | - | 16 | 16 | 13 | - | 6 | 10 |

*Table A.15 DR1 due dates and weights.*

| Job | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Due date $d_j$ | 19 | 18 | 23 | 19 | 6 | 18 | 20 | 17 | 19 | 18 |
| Weight $W_j$ | 2 | 3 | 1 | 1 | 4 | 1 | 1 | 1 | 3 | 2 |

*Table A.16 DR2 5 x 12 x 3 with 25 operations.*

| Jobs | | Machine 1 | | | Machine 2 | | | Machine 3 | | | Machine 4 | | | Machine 5 | | | Machine 6 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $j$ | 19.76 | u1 | u2 | u3 | u1 | u2 | u3 | u1 | u2 | u3 | u1 | u2 | u3 | u1 | u2 | u3 | u1 | u2 | u3 |
| 1 | 1,1 | 8 | 7 | - | 3 | 9 | 2 | 7 | 2 | 8 | 4 | 3 | - | 5 | 5 | 2 | 8 | 10 | 5 |
| | 1,2 | 5 | 7 | 2 | - | 5 | 6 | - | 10 | - | 10 | - | - | 2 | 7 | - | 6 | - | 2 |
| | 1,3 | - | - | - | - | 10 | - | 10 | 4 | - | - | - | 10 | 6 | - | - | 3 | - | 6 |
| | 1,4 | 4 | - | 9 | - | 10 | - | 6 | - | - | - | - | - | - | - | 4 | - | 5 | - |
| | 1,5 | - | 7 | - | 10 | - | 3 | 6 | 10 | - | 4 | 3 | 9 | - | - | - | 8 | 8 | 6 |
| | 1,6 | 2 | 8 | 10 | 5 | 4 | - | - | 7 | 4 | - | - | 7 | - | - | 3 | 5 | - | 2 |
| 2 | 2,1 | - | 6 | - | 2 | 9 | - | 2 | 4 | - | 2 | 5 | - | 9 | - | 5 | 10 | 6 | - |
| | 2,2 | - | 3 | - | 6 | - | - | 10 | - | 9 | - | 5 | 3 | 3 | - | - | 4 | 5 | 8 |
| | 2,3 | 4 | - | 5 | - | - | 4 | 9 | 4 | 6 | - | 9 | - | - | - | - | 10 | - | 2 |
| | 2,4 | - | 8 | 8 | 6 | - | - | - | 4 | 8 | 7 | - | 3 | 9 | 2 | 7 | 2 | 8 | 4 |
| 3 | 3,1 | 3 | 5 | - | 2 | - | - | - | 8 | 5 | 7 | 2 | - | 5 | 6 | - | 10 | - | 10 |
| | 3,2 | 5 | 10 | 6 | - | - | - | 8 | 3 | - | - | - | - | 10 | - | 10 | 4 | - | - |
| | 3,3 | - | 4 | 5 | 8 | 9 | - | - | 7 | 4 | - | 9 | - | 10 | - | 6 | - | - | - |
| | 3,4 | - | 10 | - | 2 | 3 | - | 6 | - | - | 7 | - | 10 | - | 3 | 6 | 10 | - | 4 |
| | 3,5 | 7 | 2 | 8 | 4 | 3 | - | 5 | 5 | 2 | 8 | 10 | 5 | 4 | - | - | 7 | 4 | - |
| 4 | 4,1 | - | 10 | - | 10 | - | - | 2 | 7 | - | 6 | - | 2 | 9 | - | 2 | 4 | - | 2 |
| | 4,2 | 10 | 4 | - | - | - | 10 | 6 | - | - | 3 | - | 6 | - | - | 10 | - | 9 | - |
| | 4,3 | 6 | - | - | - | - | - | - | - | 4 | - | 5 | - | - | 4 | 9 | 4 | 6 | - |
| | 4,4 | 6 | 10 | - | 4 | 3 | 9 | - | - | - | 8 | 8 | 6 | - | - | - | 4 | 8 | 7 |
| | 4,5 | - | 7 | 4 | - | - | 7 | - | - | 3 | 5 | - | 2 | - | - | - | 8 | 5 | 7 |
| | 4,6 | 2 | 4 | - | 2 | 5 | - | 9 | - | 5 | 10 | 6 | - | - | - | 8 | 3 | - | - |
| 5 | 5,1 | 10 | - | 9 | - | 5 | 3 | 3 | - | - | 4 | 5 | 8 | 9 | - | - | 7 | 4 | - |
| | 5,2 | 9 | 4 | 6 | - | 9 | - | - | - | - | 10 | - | 2 | 3 | - | 6 | - | - | 7 |
| | 5,3 | - | 4 | 8 | 7 | - | 3 | 9 | 2 | 7 | 2 | 8 | 4 | 3 | - | 5 | 5 | 2 | 8 |
| | 5,4 | - | 8 | 5 | 7 | 2 | - | 5 | 6 | - | 10 | - | 10 | - | - | 2 | 7 | - | 6 |

*Table A.17 DR2 5 x 12 x 3 with 25 operations (continued).*

| Jobs | | Machine 7 | | | Machine 8 | | | Machine 9 | | | Machine 10 | | | Machine 11 | | | Machine 12 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $j$ | 19.76 | u1 | u2 | u3 | u1 | u2 | u3 | u1 | u2 | u3 | u1 | u2 | u3 | u1 | u2 | u3 | u1 | u2 | u3 |
| 1 | 1,1 | 4 | - | - | 7 | 4 | - | - | 7 | - | - | 3 | 5 | - | 2 | - | - | - | 8 |
| | 1,2 | 9 | - | 2 | 4 | - | 2 | 5 | - | 9 | - | 5 | 10 | 6 | - | - | - | 8 | 3 |
| | 1,3 | - | - | 10 | - | 9 | - | 5 | 3 | 3 | - | - | 4 | 5 | 8 | 9 | - | - | 7 |
| | 1,4 | - | 4 | 9 | 4 | 6 | - | 9 | - | - | - | - | 10 | - | 2 | 3 | - | 6 | - |
| | 1,5 | - | - | - | 4 | 8 | 7 | - | 3 | 9 | 2 | 7 | 2 | 8 | 4 | 3 | - | 5 | 5 |
| | 1,6 | - | - | - | 8 | 5 | 7 | 2 | - | 5 | 6 | - | 10 | - | 10 | - | - | 2 | 7 |
| 2 | 2,1 | - | - | 8 | 3 | - | - | - | - | 10 | - | 10 | 4 | - | - | - | 10 | 6 | - |
| | 2,2 | 9 | - | - | 7 | 4 | - | 9 | - | 10 | - | 6 | - | - | - | - | - | - | - |
| | 2,3 | 3 | - | 6 | - | - | 7 | - | 10 | - | 3 | 6 | 10 | - | 4 | 3 | 9 | - | - |
| | 2,4 | 3 | - | 5 | 5 | 2 | 8 | 10 | 5 | 4 | - | - | 7 | 4 | - | - | 7 | - | - |
| 3 | 3,1 | - | - | 2 | 7 | - | 6 | - | 2 | 9 | - | 2 | 4 | - | 2 | 5 | - | 9 | - |
| | 3,2 | - | 10 | 6 | - | - | 3 | - | 6 | - | - | 10 | - | 9 | - | 5 | 3 | 3 | - |
| | 3,3 | - | - | - | - | 4 | - | 5 | - | - | 4 | 9 | 4 | 6 | - | 9 | - | - | - |
| | 3,4 | 3 | 9 | - | - | - | 8 | 8 | 6 | - | - | - | 4 | 8 | 7 | - | 3 | 9 | 2 |
| | 3,5 | - | 7 | - | - | 3 | 5 | - | 2 | - | - | - | 8 | 5 | 7 | 2 | - | 5 | 6 |
| 4 | 4,1 | 5 | - | 9 | - | 5 | 10 | 6 | - | - | - | 8 | 3 | - | - | - | - | 10 | - |
| | 4,2 | 5 | 3 | 3 | - | - | 4 | 5 | 8 | 9 | - | - | 7 | 4 | - | 9 | - | 10 | - |
| | 4,3 | 9 | - | - | - | - | 10 | - | 2 | 3 | - | 6 | - | - | 7 | - | 10 | - | 3 |
| | 4,4 | - | 3 | 9 | 2 | 7 | 2 | 8 | 4 | 3 | - | 5 | 5 | 2 | 8 | 10 | 5 | 4 | - |
| | 4,5 | 2 | - | 5 | 6 | - | 10 | - | 10 | - | - | 2 | 7 | - | 6 | - | 2 | 9 | - |
| | 4,6 | - | - | 10 | - | 10 | 4 | - | - | - | 10 | 6 | - | - | 3 | - | 6 | - | - |
| 5 | 5,1 | 9 | - | 10 | - | 6 | - | - | - | - | - | - | - | 4 | - | 5 | - | - | 4 |
| | 5,2 | - | 10 | - | 3 | 6 | 10 | - | 4 | 3 | 9 | - | - | - | 8 | 8 | 6 | - | - |
| | 5,3 | 10 | 5 | 4 | - | - | 7 | 4 | - | - | 7 | - | - | 3 | 5 | - | 2 | - | - |
| | 5,4 | - | 2 | 9 | - | 2 | 4 | - | 2 | 5 | - | 9 | - | 5 | 10 | 6 | - | - | - |

*Table A.18 DR2 due dates and weights.*

| Job | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Due date $d_j$ | 20 | 6 | 19 | 27 | 17 |
| Weight $W_j$ | 6 | 3 | 3 | 5 | 1 |

*Table A.19 DR3 8 x 13 x 4 with 44 operations.*

| Jobs | | Machine 1 | | | | Machine 2 | | | | Machine 3 | | | | Machine 4 | | | | Machine 5 | | | | Machine 6 | | | | Machine 7 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *j* | 23.64 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 |
| 1 | 1,1 | 12 | 14 | 10 | 3 | 11 | - | - | 4 | - | 6 | 7 | - | 12 | 9 | 13 | - | 5 | - | - | - | - | 5 | - | - | 9 | - | - | 4 |
| | 1,2 | 10 | 11 | 6 | 15 | 5 | - | 15 | - | - | - | - | 4 | - | - | - | 13 | - | - | - | - | 11 | - | - | - | - | - | - | 14 |
| | 1,3 | - | 5 | 7 | 11 | 13 | - | - | 10 | 5 | - | 12 | - | - | 15 | 8 | - | - | - | - | - | - | - | 5 | - | - | - | - | - |
| | 1,4 | - | 14 | - | 5 | 3 | 13 | - | - | - | 10 | 10 | 8 | - | - | 9 | - | 10 | 9 | 9 | - | 12 | 14 | 10 | 3 | 11 | - | - | 4 |
| | 1,5 | 14 | - | 10 | 6 | 10 | 14 | 13 | - | 8 | - | 13 | - | 14 | - | - | 15 | 9 | - | - | - | 10 | 11 | 6 | 15 | 5 | - | 15 | - |
| | 1,6 | - | - | 14 | 8 | - | - | 4 | - | - | - | - | - | 13 | - | - | 6 | 6 | 3 | 3 | - | - | 5 | 7 | 11 | 13 | - | - | 10 |
| 2 | 2,1 | 8 | - | 12 | - | - | - | - | 13 | - | 15 | 3 | - | 7 | - | - | 9 | 11 | 15 | 4 | - | - | 14 | - | 5 | 3 | 13 | - | - |
| | 2,2 | 12 | 9 | 13 | - | 5 | - | - | - | - | 5 | - | - | 9 | - | - | 4 | - | - | - | 15 | 14 | - | 10 | 6 | 10 | 14 | 13 | - |
| | 2,3 | - | - | - | 13 | - | - | - | - | 11 | - | - | - | - | - | - | 14 | - | 14 | 5 | - | - | - | 14 | 8 | - | - | 4 | - |
| | 2,4 | - | 15 | 8 | - | - | - | - | - | - | - | 5 | - | - | - | - | - | 5 | 13 | - | - | 8 | - | 12 | - | - | - | - | 13 |
| | 2,5 | - | - | 9 | - | 10 | 9 | 9 | - | 12 | 14 | 10 | 3 | 11 | - | - | 4 | - | 6 | 7 | - | 12 | 9 | 13 | - | 5 | - | - | - |
| | 2,6 | 14 | - | - | 15 | 9 | - | - | - | 10 | 11 | 6 | 15 | 5 | - | 15 | - | - | - | - | 4 | - | - | - | 13 | - | - | - | - |
| 3 | 3,1 | 13 | - | - | 6 | 6 | 3 | 3 | - | - | 5 | 7 | 11 | 13 | - | - | 10 | 5 | - | 12 | - | - | 15 | 8 | - | - | - | - | - |
| | 3,2 | 7 | - | - | 9 | 11 | 15 | 4 | - | - | 14 | - | 5 | 3 | 13 | - | - | - | 10 | 10 | 8 | - | - | 9 | - | 10 | 9 | 9 | - |
| | 3,3 | 9 | - | - | 4 | - | - | - | 15 | 14 | - | 10 | 6 | 10 | 14 | 13 | - | 8 | - | 13 | - | 14 | - | - | 15 | 9 | - | - | - |
| | 3,4 | - | - | - | 14 | - | 14 | 5 | - | - | - | 14 | 8 | - | - | 4 | - | - | - | - | - | 13 | - | - | 6 | 6 | 3 | 3 | - |
| | 3,5 | - | - | - | - | 5 | 13 | - | - | 8 | - | 12 | - | - | - | - | 13 | - | 15 | 3 | - | 7 | - | - | 9 | 11 | 15 | 4 | - |
| | 3,6 | 11 | - | - | 4 | - | 6 | 7 | - | 12 | 9 | 13 | - | 5 | - | - | - | - | 5 | - | - | 9 | - | - | 4 | - | - | - | 15 |
| 4 | 4,1 | 5 | - | 15 | - | - | - | - | 4 | - | - | - | 13 | - | - | - | - | 11 | - | - | - | - | - | - | 14 | - | 14 | 5 | - |
| | 4,2 | 13 | - | - | 10 | 5 | - | 12 | - | - | 15 | 8 | - | - | - | - | - | - | - | 5 | - | - | - | - | - | 5 | 13 | - | - |
| | 4,3 | 3 | 13 | - | - | - | 10 | 10 | 8 | - | - | 9 | - | 10 | 9 | 9 | - | 12 | 14 | 10 | 3 | 11 | - | - | 4 | - | 6 | 7 | - |
| | 4,4 | 10 | 14 | 13 | - | 8 | - | 13 | - | 14 | - | - | 15 | 9 | - | - | - | 10 | 11 | 6 | 15 | 5 | - | 15 | - | - | - | - | 4 |
| | 4,5 | - | - | 4 | - | - | - | - | - | 13 | - | - | 6 | 6 | 3 | 3 | - | - | 5 | 7 | 11 | 13 | - | - | 10 | 5 | - | 12 | - |
| | 4,6 | - | - | - | 13 | - | 15 | 3 | - | 7 | - | - | 9 | 11 | 15 | 4 | - | - | 14 | - | 5 | 3 | 13 | - | - | - | 10 | 10 | 8 |
| 5 | 5,1 | 5 | - | - | - | - | 5 | - | - | 9 | - | - | 4 | - | - | - | 15 | 14 | - | 10 | 6 | 10 | 14 | 13 | - | 8 | - | 13 | - |
| | 5,2 | - | - | - | - | 11 | - | - | - | - | - | - | 14 | - | 14 | 5 | - | - | - | 14 | 8 | - | - | 4 | - | - | - | - | - |
| | 5,3 | - | - | - | - | - | - | 5 | - | - | - | - | - | 5 | 13 | - | - | 8 | - | 12 | - | - | - | - | 13 | - | 15 | 3 | - |
| | 5,4 | 10 | 9 | 9 | - | 12 | 14 | 10 | 3 | 11 | - | - | 4 | - | 6 | 7 | - | 12 | 9 | 13 | - | 5 | - | - | - | - | 5 | - | - |

*Table A.20 DR3 8 x 13 x 4 with 44 operations (continued).*

| Jobs | | Machine 8 | | | | Machine 9 | | | | Machine 10 | | | | Machine 11 | | | | Machine 12 | | | | Machine 13 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *j* | 23.64 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 |
| 1 | 1,1 | - | - | - | 15 | 14 | - | 10 | 6 | 10 | 14 | 13 | - | 8 | - | 13 | - | 14 | - | - | 15 | 9 | - | - | - |
| | 1,2 | - | 14 | 5 | - | - | - | 14 | 8 | - | - | 4 | - | - | - | - | - | 13 | - | - | 6 | 6 | 3 | 3 | - |
| | 1,3 | 5 | 13 | - | - | 8 | - | 12 | - | - | - | - | 13 | - | 15 | 3 | - | 7 | - | - | 9 | 11 | 15 | 4 | - |
| | 1,4 | - | 6 | 7 | - | 12 | 9 | 13 | - | 5 | - | - | - | - | 5 | - | - | 9 | - | - | 4 | - | - | - | 15 |
| | 1,5 | - | - | - | 4 | - | - | - | 13 | - | - | - | - | 11 | - | - | - | - | - | - | 14 | - | 14 | 5 | - |
| | 1,6 | 5 | - | 12 | - | - | 15 | 8 | - | - | - | - | - | - | - | 5 | - | - | - | - | - | 5 | 13 | - | - |
| 2 | 2,1 | - | 10 | 10 | 8 | - | - | 9 | - | 10 | 9 | 9 | - | 12 | 14 | 10 | 3 | 11 | - | - | 4 | - | 6 | 7 | - |
| | 2,2 | 8 | - | 13 | - | 14 | - | - | 15 | 9 | - | - | - | 10 | 11 | 6 | 15 | 5 | - | 15 | - | - | - | - | 4 |
| | 2,3 | - | - | - | - | 13 | - | - | 6 | 6 | 3 | 3 | - | - | 5 | 7 | 11 | 13 | - | - | 10 | 5 | - | 12 | - |
| | 2,4 | - | 15 | 3 | - | 7 | - | - | 9 | 11 | 15 | 4 | - | - | 14 | - | 5 | 3 | 13 | - | - | - | 10 | 10 | 8 |
| | 2,5 | - | 5 | - | - | 9 | - | - | 4 | - | - | - | 15 | 14 | - | 10 | 6 | 10 | 14 | 13 | - | 8 | - | 13 | - |
| | 2,6 | 11 | - | - | - | - | - | - | 14 | - | 14 | 5 | - | - | - | 14 | 8 | - | - | 4 | - | - | - | - | - |
| 3 | 3,1 | - | - | 5 | - | - | - | - | - | 5 | 13 | - | - | 8 | - | 12 | - | - | - | - | 13 | - | 15 | 3 | - |
| | 3,2 | 12 | 14 | 10 | 3 | 11 | - | - | 4 | - | 6 | 7 | - | 12 | 9 | 13 | - | 5 | - | - | - | - | 5 | - | - |
| | 3,3 | 10 | 11 | 6 | 15 | 5 | - | 15 | - | - | - | - | 4 | - | - | - | 13 | - | - | - | - | 11 | - | - | - |
| | 3,4 | - | 5 | 7 | 11 | 13 | - | - | 10 | 5 | - | 12 | - | - | 15 | 8 | - | - | - | - | - | - | - | 5 | - |
| | 3,5 | - | 14 | - | 5 | 3 | 13 | - | - | - | 10 | 10 | 8 | - | - | 9 | - | 10 | 9 | 9 | - | 12 | 14 | 10 | 3 |
| | 3,6 | 14 | - | 10 | 6 | 10 | 14 | 13 | - | 8 | - | 13 | - | 14 | - | - | 15 | 9 | - | - | - | 10 | 11 | 6 | 15 |
| 4 | 4,1 | - | - | 14 | 8 | - | - | 4 | - | - | - | - | - | 13 | - | - | 6 | 6 | 3 | 3 | - | - | 5 | 7 | 11 |
| | 4,2 | 8 | - | 12 | - | - | - | - | 13 | - | 15 | 3 | - | 7 | - | - | 9 | 11 | 15 | 4 | - | - | 14 | - | 5 |
| | 4,3 | 12 | 9 | 13 | - | 5 | - | - | - | - | 5 | - | - | 9 | - | - | 4 | - | - | - | 15 | 14 | - | 10 | 6 |
| | 4,4 | - | - | - | 13 | - | - | - | - | 11 | - | - | - | - | - | - | 14 | - | 14 | 5 | - | - | - | 14 | 8 |
| | 4,5 | - | 15 | 8 | - | - | - | - | - | - | - | 5 | - | - | - | - | - | 5 | 13 | - | - | 8 | - | 12 | - |
| | 4,6 | - | - | 9 | - | 10 | 9 | 9 | - | 12 | 14 | 10 | 3 | 11 | - | - | 4 | - | 6 | 7 | - | 12 | 9 | 13 | - |
| 5 | 5,1 | 14 | - | - | 15 | 9 | - | - | - | 10 | 11 | 6 | 15 | 5 | - | 15 | - | - | - | - | 4 | - | - | - | 13 |
| | 5,2 | 13 | - | - | 6 | 6 | 3 | 3 | - | - | 5 | 7 | 11 | 13 | - | - | 10 | 5 | - | 12 | - | - | 15 | 8 | - |
| | 5,3 | 7 | - | - | 9 | 11 | 15 | 4 | - | - | 14 | - | 5 | 3 | 13 | - | - | - | 10 | 10 | 8 | - | - | 9 | - |
| | 5,4 | 9 | - | - | 4 | - | - | - | 15 | 14 | - | 10 | 6 | 10 | 14 | 13 | - | 8 | - | 13 | - | 14 | - | - | 15 |

*Table A.21 DR3 8 x 13 x 4 with 44 operations (continued).*

| Jobs | | Machine 1 | | | | Machine 2 | | | | Machine 3 | | | | Machine 4 | | | | Machine 5 | | | | Machine 6 | | | | Machine 7 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $j$ | 23.64 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 |
| 6 | 6,1 | 9 | - | - | - | 10 | 11 | 6 | 15 | 5 | - | 15 | - | - | - | - | 4 | - | - | - | 13 | - | - | - | - | 11 | - | - | - |
| | 6,2 | 6 | 3 | 3 | - | - | 5 | 7 | 11 | 13 | - | - | 10 | 5 | - | 12 | - | - | 15 | 8 | - | - | - | - | - | - | - | 5 | - |
| | 6,3 | 11 | 15 | 4 | - | - | 14 | - | 5 | 3 | 13 | - | - | - | 10 | 10 | 8 | - | - | 9 | - | 10 | 9 | 9 | - | 12 | 14 | 10 | 3 |
| | 6,4 | - | - | - | 15 | 14 | - | 10 | 6 | 10 | 14 | 13 | - | 8 | - | 13 | - | 14 | - | - | 15 | 9 | - | - | - | 10 | 11 | 6 | 15 |
| | 6,5 | - | 14 | 5 | - | - | - | 14 | 8 | - | - | 4 | - | - | - | - | - | 13 | - | - | 6 | 6 | 3 | 3 | - | - | 5 | 7 | 11 |
| 7 | 7,1 | 5 | 13 | - | - | 8 | - | 12 | - | - | - | - | 13 | - | 15 | 3 | - | 7 | - | - | 9 | 11 | 15 | 4 | - | - | 14 | - | 5 |
| | 7,2 | - | 6 | 7 | - | 12 | 9 | 13 | - | 5 | - | - | - | - | 5 | - | - | 9 | - | - | 4 | - | - | - | 15 | 14 | - | 10 | 6 |
| | 7,3 | - | - | - | 4 | - | - | - | 13 | - | - | - | - | 11 | - | - | - | - | - | - | 14 | - | 14 | 5 | - | - | - | 14 | 8 |
| | 7,4 | 5 | - | 12 | - | - | 15 | 8 | - | - | - | - | - | - | - | 5 | - | - | - | - | - | 5 | 13 | - | - | 8 | - | 12 | - |
| | 7,5 | - | 10 | 10 | 8 | - | - | 9 | - | 10 | 9 | 9 | - | 12 | 14 | 10 | 3 | 11 | - | - | 4 | - | 6 | 7 | - | 12 | 9 | 13 | - |
| 8 | 8,1 | 8 | - | 13 | - | 14 | - | - | 15 | 9 | - | - | - | 10 | 11 | 6 | 15 | 5 | - | 15 | - | - | - | - | 4 | - | - | - | 13 |
| | 8,2 | - | - | - | - | 13 | - | - | 6 | 6 | 3 | 3 | - | - | 5 | 7 | 11 | 13 | - | - | 10 | 5 | - | 12 | - | - | 15 | 8 | - |
| | 8,3 | - | 15 | 3 | - | 7 | - | - | 9 | 11 | 15 | 4 | - | - | 14 | - | 5 | 3 | 13 | - | - | - | 10 | 10 | 8 | - | - | 9 | - |
| | 8,4 | - | 5 | - | - | 9 | - | - | 4 | - | - | - | 15 | 14 | - | 10 | 6 | 10 | 14 | 13 | - | 8 | - | 13 | - | 14 | - | - | 15 |
| | 8,5 | 11 | - | - | - | - | - | - | 14 | - | 14 | 5 | - | - | - | 14 | 8 | - | - | 4 | - | - | - | - | - | 13 | - | - | 6 |
| | 8,6 | - | - | 5 | - | - | - | - | - | 5 | 13 | - | - | 8 | - | 12 | - | - | - | - | 13 | - | 15 | 3 | - | 7 | - | - | 9 |

*Table A.22 DR3 8 x 13 x 4 with 44 operations (continued).*

| Jobs | | Machine 8 | | | | Machine 9 | | | | Machine 10 | | | | Machine 11 | | | | Machine 12 | | | | Machine 13 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $j$ | 23.64 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 | u1 | u2 | u3 | u4 |
| | 6,1 | - | - | - | 14 | - | 14 | 5 | - | - | - | 14 | 8 | - | - | 4 | - | - | - | - | - | 13 | - | - | 6 |
| | 6,2 | - | - | - | - | 5 | 13 | - | - | 8 | - | 12 | - | - | - | - | 13 | - | 15 | 3 | - | 7 | - | - | 9 |
| 6 | 6,3 | 11 | - | - | 4 | - | 6 | 7 | - | 12 | 9 | 13 | - | 5 | - | - | - | - | 5 | - | - | 9 | - | - | 4 |
| | 6,4 | 5 | - | 15 | - | - | - | - | 4 | - | - | - | 13 | - | - | - | - | 11 | - | - | - | - | - | - | 14 |
| | 6,5 | 13 | - | - | 10 | 5 | - | 12 | - | - | 15 | 8 | - | - | - | - | - | - | - | 5 | - | - | - | - | - |
| | 7,1 | 3 | 13 | - | - | - | 10 | 10 | 8 | - | - | 9 | - | 10 | 9 | 9 | - | 12 | 14 | 10 | 3 | 11 | - | - | 4 |
| | 7,2 | 10 | 14 | 13 | - | 8 | - | 13 | - | 14 | - | - | 15 | 9 | - | - | - | 10 | 11 | 6 | 15 | 5 | - | 15 | - |
| 7 | 7,3 | - | - | 4 | - | - | - | - | - | 13 | - | - | 6 | 6 | 3 | 3 | - | - | 5 | 7 | 11 | 13 | - | - | 10 |
| | 7,4 | - | - | - | 13 | - | 15 | 3 | - | 7 | - | - | 9 | 11 | 15 | 4 | - | - | 14 | - | 5 | 3 | 13 | - | - |
| | 7,5 | 5 | - | - | - | - | 5 | - | - | 9 | - | - | 4 | - | - | - | 15 | 14 | - | 10 | 6 | 10 | 14 | 13 | - |
| | 8,1 | - | - | - | - | 11 | - | - | - | - | - | - | 14 | - | 14 | 5 | - | - | - | 14 | 8 | - | - | 4 | - |
| | 8,2 | - | - | - | - | - | - | 5 | - | - | - | - | - | 5 | 13 | - | - | 8 | - | 12 | - | - | - | - | 13 |
| | 8,3 | 10 | 9 | 9 | - | 12 | 14 | 10 | 3 | 11 | - | - | 4 | - | 6 | 7 | - | 12 | 9 | 13 | - | 5 | - | - | - |
| 8 | 8,4 | 9 | - | - | - | 10 | 11 | 6 | 15 | 5 | - | 15 | - | - | - | - | 4 | - | - | - | 13 | - | - | - | - |
| | 8,5 | 6 | 3 | 3 | - | - | 5 | 7 | 11 | 13 | - | - | 10 | 5 | - | 12 | - | - | 15 | 8 | - | - | - | - | - |
| | 8,6 | 11 | 15 | 4 | - | - | 14 | - | 5 | 3 | 13 | - | - | - | 10 | 10 | 8 | - | - | 9 | - | 10 | 9 | 9 | - |

*Table A.23 DR3 due dates and weights.*

| Job | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Due date $d_j$ | 19 | 35 | 12 | 28 | 15 | 20 | 22 | 31 |
| Weight $W_j$ | 2 | 2 | 4 | 5 | 1 | 5 | 4 | 4 |

*Table A.24 DR4 6 x 6 x 3 with 41 operations.*

| Jobs | | Machine 1 | | | Machine 2 | | | Machine 3 | | | Machine 4 | | | Machine 5 | | | Machine 6 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $j$ | 8.83 | u1 | u2 | u3 | u1 | u2 | u3 | u1 | u2 | u3 | u1 | u2 | u3 | u1 | u2 | u3 | u1 | u2 | u3 |
| 1 | 1,1 | - | - | 8 | - | 8 | 5 | - | - | - | 8 | 6 | - | - | 4 | - | - | - | - |
| | 1,2 | - | 8 | - | 7 | - | 5 | 4 | 4 | - | - | 5 | 5 | 7 | 8 | - | - | 7 | 5 |
| | 1,3 | - | 7 | - | - | 4 | 6 | - | - | - | - | - | - | - | 5 | - | 5 | - | - |
| | 1,4 | 5 | 8 | - | - | 6 | - | 7 | - | - | - | - | 8 | - | 4 | 4 | - | 6 | - |
| | 1,5 | - | 6 | 7 | 8 | 4 | - | - | 8 | - | 5 | 4 | 8 | - | - | - | 7 | 7 | 6 |
| | 1,6 | - | - | - | 5 | 7 | 6 | - | 4 | 7 | 8 | 6 | 4 | 7 | - | - | 4 | - | 5 |
| | 1,7 | 5 | - | 7 | 6 | 8 | - | 4 | - | - | - | 8 | - | - | - | 6 | - | - | 4 |
| | 1,8 | 5 | - | 8 | 8 | - | 7 | 5 | 6 | 8 | - | 5 | 6 | - | 8 | - | 8 | - | - |
| 2 | 2,1 | 4 | 6 | - | - | - | 7 | 7 | 5 | 4 | 5 | - | 4 | 5 | - | 8 | - | 5 | - |
| | 2,2 | 8 | - | - | - | - | 7 | 4 | - | - | - | - | 8 | - | 8 | 5 | - | - | - |
| | 2,3 | 8 | 6 | - | - | 4 | - | - | - | - | - | 8 | - | 7 | - | 5 | 4 | 4 | - |
| | 2,4 | - | 5 | 5 | 7 | 8 | - | - | 7 | 5 | - | 7 | - | - | 4 | 6 | - | - | - |
| | 2,5 | - | - | - | - | 5 | - | 5 | - | - | 5 | 8 | - | - | 6 | - | 7 | - | - |
| | 2,6 | - | - | 8 | - | 4 | 4 | - | 6 | - | - | 6 | 7 | 8 | 4 | - | - | 8 | - |
| | 2,7 | 5 | 4 | 8 | - | - | - | 7 | 7 | 6 | - | - | - | 5 | 7 | 6 | - | 4 | 7 |
| | 2,8 | 8 | 6 | 4 | 7 | - | - | 4 | - | 5 | 5 | - | 7 | 6 | 8 | - | 4 | - | - |
| 3 | 3,1 | - | 8 | - | - | - | 6 | - | - | 4 | 5 | - | 8 | 8 | - | 7 | 5 | 6 | 8 |
| | 3,2 | - | 5 | 6 | - | 8 | - | 8 | - | - | 4 | 6 | - | - | - | 7 | 7 | 5 | 4 |
| | 3,3 | 5 | - | 4 | 5 | - | 8 | - | 5 | - | 8 | - | - | - | - | 7 | 4 | - | - |
| | 3,4 | - | - | 8 | - | 8 | 5 | - | - | - | 8 | 6 | - | - | 4 | - | - | - | - |
| | 3,5 | - | 8 | - | 7 | - | 5 | 4 | 4 | - | - | 5 | 5 | 7 | 8 | - | - | 7 | 5 |
| | 3,6 | - | 7 | - | - | 4 | 6 | - | - | - | - | - | - | - | 5 | - | 5 | - | - |
| | 3,7 | 5 | 8 | - | - | 6 | - | 7 | - | - | - | - | 8 | - | 4 | 4 | - | 6 | - |
| | 3,8 | - | 6 | 7 | 8 | 4 | - | - | 8 | - | 5 | 4 | 8 | - | - | - | 7 | 7 | 6 |
| 4 | 4,1 | - | - | - | 5 | 7 | 6 | - | 4 | 7 | 8 | 6 | 4 | 7 | - | - | 4 | - | 5 |
| | 4,2 | 5 | - | 7 | 6 | 8 | - | 4 | - | - | - | 8 | - | - | - | 6 | - | - | 4 |
| | 4,3 | 5 | - | 8 | 8 | - | 7 | 5 | 6 | 8 | - | 5 | 6 | - | 8 | - | 8 | - | - |
| | 4,4 | 4 | 6 | - | - | - | 7 | 7 | 5 | 4 | 5 | - | 4 | 5 | - | 8 | - | 5 | - |
| | 4,5 | 8 | - | - | - | - | 7 | 4 | - | - | - | - | 8 | - | 8 | 5 | - | - | - |
| 5 | 5,1 | 8 | 6 | - | - | 4 | - | - | - | - | - | 8 | - | 7 | - | 5 | 4 | 4 | - |
| | 5,2 | - | 5 | 5 | 7 | 8 | - | - | 7 | 5 | - | 7 | - | - | 4 | 6 | - | - | - |
| | 5,3 | - | - | - | - | 5 | - | 5 | - | - | 5 | 8 | - | - | 6 | - | 7 | - | - |
| | 5,4 | - | - | 8 | - | 4 | 4 | - | 6 | - | - | 6 | 7 | 8 | 4 | - | - | 8 | - |
| | 5,5 | 5 | 4 | 8 | - | - | - | 7 | 7 | 6 | - | - | - | 5 | 7 | 6 | - | 4 | 7 |
| | 5,6 | 8 | 6 | 4 | 7 | - | - | 4 | - | 5 | 5 | - | 7 | 6 | 8 | - | 4 | - | - |
| 6 | 6,1 | - | 8 | - | - | - | 6 | - | - | 4 | 5 | - | 8 | 8 | - | 7 | 5 | 6 | 8 |
| | 6,2 | - | 5 | 6 | - | 8 | - | 8 | - | - | 4 | 6 | - | - | - | 7 | 7 | 5 | 4 |
| | 6,3 | 5 | - | 4 | 5 | - | 8 | - | 5 | - | 8 | - | - | - | - | 7 | 4 | - | - |
| | 6,4 | - | - | 8 | - | 8 | 5 | - | - | - | 8 | 6 | - | - | 4 | - | - | - | - |
| | 6,5 | - | 8 | - | 7 | - | 5 | 4 | 4 | - | - | 5 | 5 | 7 | 8 | - | - | 7 | 5 |
| | 6,6 | - | 7 | - | - | 4 | 6 | - | - | - | - | - | - | - | 5 | - | 5 | - | - |

*Table A.25 DR4 due dates and weights.*

| Job | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Due date $d_j$ | 19 | 20 | 6 | 18 | 9 | 11 |
| Weight $W_j$ | 7 | 2 | 7 | 4 | 1 | 5 |

*Table A.26 DR5 7 x 8 x 5 with 47 operations.*

| Jobs | | Machine 1 | | | | | Machine 2 | | | | | Machine 3 | | | | | Machine 4 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *j* | 22.77 | u1 | u2 | u3 | u4 | u5 | u1 | u2 | u3 | u4 | u5 | u1 | u2 | u3 | u4 | u5 | u1 | u2 | u3 | u4 | u5 |
| | 1,1 | 9 | - | - | 9 | 12 | 13 | - | 11 | - | - | 3 | - | 5 | 6 | 14 | 10 | 14 | 5 | 3 | - |
| | 1,2 | - | 8 | - | - | 5 | - | 3 | - | 15 | 8 | 7 | 15 | 14 | 12 | - | 15 | 4 | - | 15 | - |
| | 1,3 | - | - | - | 14 | 8 | - | - | 3 | - | - | 2 | - | - | 6 | 12 | - | 5 | 2 | 2 | - |
| 1 | 1,4 | 4 | - | - | 10 | - | 4 | - | 7 | - | 7 | - | 11 | - | 3 | - | 13 | - | 15 | 2 | - |
| | 1,5 | - | 10 | 8 | 9 | - | - | 9 | 12 | 13 | - | 11 | - | - | 3 | - | 5 | 6 | 14 | 10 | 14 |
| | 1,6 | 9 | 14 | 13 | - | 8 | - | - | 5 | - | 3 | - | 15 | 8 | 7 | 15 | 14 | 12 | - | 15 | 4 |
| | 1,7 | 2 | 14 | 4 | - | - | - | 14 | 8 | - | - | 3 | - | - | 2 | - | - | 6 | 12 | - | 5 |
| | 1,8 | - | - | - | 4 | - | - | 10 | - | 4 | - | 7 | - | 7 | - | 11 | - | 3 | - | 13 | - |
| | 2,1 | 7 | 11 | 8 | - | 10 | 8 | 9 | - | - | 9 | 12 | 13 | - | 11 | - | - | 3 | - | 5 | 6 |
| | 2,2 | 15 | 9 | 6 | 9 | 14 | 13 | - | 8 | - | - | 5 | - | 3 | - | 15 | 8 | 7 | 15 | 14 | 12 |
| | 2,3 | - | - | - | 2 | 14 | 4 | - | - | - | 14 | 8 | - | - | 3 | - | - | 2 | - | - | 6 |
| 2 | 2,4 | - | 9 | 10 | - | - | - | 4 | - | - | 10 | - | 4 | - | 7 | - | 7 | - | 11 | - | 3 |
| | 2,5 | - | 10 | 10 | 7 | 11 | 8 | - | 10 | 8 | 9 | - | - | 9 | 12 | 13 | - | 11 | - | - | 3 |
| | 2,6 | 7 | - | 15 | 15 | 9 | 6 | 9 | 14 | 13 | - | 8 | - | - | 5 | - | 3 | - | 15 | 8 | 7 |
| | 2,7 | - | - | - | - | - | - | 2 | 14 | 4 | - | - | - | 14 | 8 | - | - | 3 | - | - | 2 |
| | 2,8 | - | 13 | 12 | - | 9 | 10 | - | - | - | 4 | - | - | 10 | - | 4 | - | 7 | - | 7 | - |
| | 3,1 | 12 | - | - | - | 10 | 10 | 7 | 11 | 8 | - | 10 | 8 | 9 | - | - | 9 | 12 | 13 | - | 11 |
| | 3,2 | - | 3 | - | 7 | - | 15 | 15 | 9 | 6 | 9 | 14 | 13 | - | 8 | - | - | 5 | - | 3 | - |
| 3 | 3,3 | - | 11 | - | - | - | - | - | - | - | 2 | 14 | 4 | - | - | - | 14 | 8 | - | - | 3 |
| | 3,4 | 4 | - | 11 | - | 13 | 12 | - | 9 | 10 | - | - | - | 4 | - | - | 10 | - | 4 | - | 7 |
| | 3,5 | 2 | 7 | 2 | 12 | - | - | - | 10 | 10 | 7 | 11 | 8 | - | 10 | 8 | 9 | - | - | 9 | 12 |
| | 4,1 | - | 8 | - | - | 3 | - | 7 | - | 15 | 15 | 9 | 6 | 9 | 14 | 13 | - | 8 | - | - | 5 |
| | 4,2 | 12 | - | 2 | - | 11 | - | - | - | - | - | - | - | 2 | 14 | 4 | - | - | - | 14 | 8 |
| | 4,3 | - | - | 9 | 4 | - | 11 | - | 13 | 12 | - | 9 | 10 | - | - | - | 4 | - | - | 10 | - |
| 4 | 4,4 | 3 | - | 13 | 2 | 7 | 2 | 12 | - | - | - | 10 | 10 | 7 | 11 | 8 | - | 10 | 8 | 9 | - |
| | 4,5 | 12 | - | - | - | 8 | - | - | 3 | - | 7 | - | 15 | 15 | 9 | 6 | 9 | 14 | 13 | - | 8 |
| | 4,6 | - | - | 11 | 12 | - | 2 | - | 11 | - | - | - | - | - | - | - | 2 | 14 | 4 | - | - |
| | 4,7 | 6 | 10 | 12 | - | - | 9 | 4 | - | 11 | - | 13 | 12 | - | 9 | 10 | - | - | - | 4 | - |
| | 4,8 | 8 | 10 | 15 | 3 | - | 13 | 2 | 7 | 2 | 12 | - | - | - | 10 | 10 | 7 | 11 | 8 | - | 10 |
| | 5,1 | - | - | - | 12 | - | - | - | 8 | - | - | 3 | - | 7 | - | 15 | 15 | 9 | 6 | 9 | 14 |
| | 5,2 | - | 8 | 12 | - | - | 11 | 12 | - | 2 | - | 11 | - | - | - | - | - | - | - | 2 | 14 |
| | 5,3 | - | - | 8 | 6 | 10 | 12 | - | - | 9 | 4 | - | 11 | - | 13 | 12 | - | 9 | 10 | - | - |
| 5 | 5,4 | - | 7 | 8 | 8 | 10 | 15 | 3 | - | 13 | 2 | 7 | 2 | 12 | - | - | - | 10 | 10 | 7 | 11 |
| | 5,5 | 14 | 5 | 3 | - | - | - | 12 | - | - | - | 8 | - | - | 3 | - | 7 | - | 15 | 15 | 9 |
| | 5,6 | 4 | - | 15 | - | 8 | 12 | - | - | 11 | 12 | - | 2 | - | 11 | - | - | - | - | - | - |
| | 5,7 | 5 | 2 | 2 | - | - | 8 | 6 | 10 | 12 | - | - | 9 | 4 | - | 11 | - | 13 | 12 | - | 9 |
| | 5,8 | - | 15 | 2 | - | 7 | 8 | 8 | 10 | 15 | 3 | - | 13 | 2 | 7 | 2 | 12 | - | - | - | 10 |
| | 6,1 | 6 | 14 | 10 | 14 | 5 | 3 | - | - | - | 12 | - | - | - | 8 | - | - | 3 | - | 7 | - |
| | 6,2 | 12 | - | 15 | 4 | - | 15 | - | 8 | 12 | - | - | 11 | 12 | - | 2 | - | 11 | - | - | - |
| 6 | 6,3 | 6 | 12 | - | 5 | 2 | 2 | - | - | 8 | 6 | 10 | 12 | - | - | 9 | 4 | - | 11 | - | 13 |
| | 6,4 | 3 | - | 13 | - | 15 | 2 | - | 7 | 8 | 8 | 10 | 15 | 3 | - | 13 | 2 | 7 | 2 | 12 | - |
| | 6,5 | 3 | - | 5 | 6 | 14 | 10 | 14 | 5 | 3 | - | - | - | 12 | - | - | - | 8 | - | - | 3 |
| | 7,1 | 7 | 15 | 14 | 12 | - | 15 | 4 | - | 15 | - | 8 | 12 | - | - | 11 | 12 | - | 2 | - | 11 |
| | 7,2 | 2 | - | - | 6 | 12 | - | 5 | 2 | 2 | - | - | 8 | 6 | 10 | 12 | - | - | 9 | 4 | - |
| 7 | 7,3 | - | 11 | - | 3 | - | 13 | - | 15 | 2 | - | 7 | 8 | 8 | 10 | 15 | 3 | - | 13 | 2 | 7 |
| | 7,4 | 11 | - | - | 3 | - | 5 | 6 | 14 | 10 | 14 | 5 | 3 | - | - | - | 12 | - | - | - | 8 |
| | 7,5 | - | 15 | 8 | 7 | 15 | 14 | 12 | - | 15 | 4 | - | 15 | - | 8 | 12 | - | - | 11 | 12 | - |

*Table A.27 DR5 7 x 8 x 5 with 47 operations (continued).*

| Job | | Machine 5 | | | | | Machine 6 | | | | | Machine 7 | | | | | Machine 8 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $j$ | 22.77 | u1 | u2 | u3 | u4 | u5 | u1 | u2 | u3 | u4 | u5 | u1 | u2 | u3 | u4 | u5 | u1 | u2 | u3 | u4 | u5 |
| 1 | 1,1 | - | - | 12 | - | - | - | 8 | - | - | 3 | - | 7 | - | 15 | 15 | 9 | 6 | 9 | 14 | 13 |
| | 1,2 | 8 | 12 | - | - | 11 | 12 | - | 2 | - | 11 | - | - | - | - | - | - | - | 2 | 14 | 4 |
| | 1,3 | - | 8 | 6 | 10 | 12 | - | - | 9 | 4 | - | 11 | - | 13 | 12 | - | 9 | 10 | - | - | - |
| | 1,4 | 7 | 8 | 8 | 10 | 15 | 3 | - | 13 | 2 | 7 | 2 | 12 | - | - | - | 10 | 10 | 7 | 11 | 8 |
| | 1,5 | 5 | 3 | - | - | - | 12 | - | - | - | 8 | - | - | 3 | - | 7 | - | 15 | 15 | 9 | 6 |
| | 1,6 | - | 15 | - | 8 | 12 | - | - | 11 | 12 | - | 2 | - | 11 | - | - | - | - | - | - | - |
| | 1,7 | 2 | 2 | - | - | 8 | 6 | 10 | 12 | - | - | 9 | 4 | - | 11 | - | 13 | 12 | - | 9 | 10 |
| | 1,8 | 15 | 2 | - | 7 | 8 | 8 | 10 | 15 | 3 | - | 13 | 2 | 7 | 2 | 12 | - | - | - | 10 | 10 |
| 2 | 2,1 | 14 | 10 | 14 | 5 | 3 | - | - | - | 12 | - | - | - | 8 | - | - | 3 | - | 7 | - | 15 |
| | 2,2 | - | 15 | 4 | - | 15 | - | 8 | 12 | - | - | 11 | 12 | - | 2 | - | 11 | - | - | - | - |
| | 2,3 | 12 | - | 5 | 2 | 2 | - | - | 8 | 6 | 10 | 12 | - | - | 9 | 4 | - | 11 | - | 13 | 12 |
| | 2,4 | - | 13 | - | 15 | 2 | - | 7 | 8 | 8 | 10 | 15 | 3 | - | 13 | 2 | 7 | 2 | 12 | - | - |
| | 2,5 | - | 5 | 6 | 14 | 10 | 14 | 5 | 3 | - | - | - | 12 | - | - | - | 8 | - | - | 3 | - |
| | 2,6 | 15 | 14 | 12 | - | 15 | 4 | - | 15 | - | 8 | 12 | - | - | 11 | 12 | - | 2 | - | 11 | - |
| | 2,7 | - | - | 6 | 12 | - | 5 | 2 | 2 | - | - | 8 | 6 | 10 | 12 | - | - | 9 | 4 | - | 11 |
| | 2,8 | 11 | - | 3 | - | 13 | - | 15 | 2 | - | 7 | 8 | 8 | 10 | 15 | 3 | - | 13 | 2 | 7 | 2 |
| 3 | 3,1 | - | - | 3 | - | 5 | 6 | 14 | 10 | 14 | 5 | 3 | - | - | - | 12 | - | - | - | 8 | - |
| | 3,2 | 15 | 8 | 7 | 15 | 14 | 12 | - | 15 | 4 | - | 15 | - | 8 | 12 | - | - | 11 | 12 | - | 2 |
| | 3,3 | - | - | 2 | - | - | 6 | 12 | - | 5 | 2 | 2 | - | - | 8 | 6 | 10 | 12 | - | - | 9 |
| | 3,4 | - | 7 | - | 11 | - | 3 | - | 13 | - | 15 | 2 | - | 7 | 8 | 8 | 10 | 15 | 3 | - | 13 |
| | 3,5 | 13 | - | 11 | - | - | 3 | - | 5 | 6 | 14 | 10 | 14 | 5 | 3 | - | - | - | 12 | - | - |
| 4 | 4,1 | - | 3 | - | 15 | 8 | 7 | 15 | 14 | 12 | - | 15 | 4 | - | 15 | - | 8 | 12 | - | - | 11 |
| | 4,2 | - | - | 3 | - | - | 2 | - | - | 6 | 12 | - | 5 | 2 | 2 | - | - | 8 | 6 | 10 | 12 |
| | 4,3 | 4 | - | 7 | - | 7 | - | 11 | - | 3 | - | 13 | - | 15 | 2 | - | 7 | 8 | 8 | 10 | 15 |
| | 4,4 | - | 9 | 12 | 13 | - | 11 | - | - | 3 | - | 5 | 6 | 14 | 10 | 14 | 5 | 3 | - | - | - |
| | 4,5 | - | - | 5 | - | 3 | - | 15 | 8 | 7 | 15 | 14 | 12 | - | 15 | 4 | - | 15 | - | 8 | 12 |
| | 4,6 | - | 14 | 8 | - | - | 3 | - | - | 2 | - | - | 6 | 12 | - | 5 | 2 | 2 | - | - | 8 |
| | 4,7 | - | 10 | - | 4 | - | 7 | - | 7 | - | 11 | - | 3 | - | 13 | - | 15 | 2 | - | 7 | 8 |
| | 4,8 | 8 | 9 | - | - | 9 | 12 | 13 | - | 11 | - | - | 3 | - | 5 | 6 | 14 | 10 | 14 | 5 | 3 |
| 5 | 5,1 | 13 | - | 8 | - | - | 5 | - | 3 | - | 15 | 8 | 7 | 15 | 14 | 12 | - | 15 | 4 | - | 15 |
| | 5,2 | 4 | - | - | - | 14 | 8 | - | - | 3 | - | - | 2 | - | - | 6 | 12 | - | 5 | 2 | 2 |
| | 5,3 | - | 4 | - | - | 10 | - | 4 | - | 7 | - | 7 | - | 11 | - | 3 | - | 13 | - | 15 | 2 |
| | 5,4 | 8 | - | 10 | 8 | 9 | - | - | 9 | 12 | 13 | - | 11 | - | - | 3 | - | 5 | 6 | 14 | 10 |
| | 5,5 | 6 | 9 | 14 | 13 | - | 8 | - | - | 5 | - | 3 | - | 15 | 8 | 7 | 15 | 14 | 12 | - | 15 |
| | 5,6 | - | 2 | 14 | 4 | - | - | - | 14 | 8 | - | - | 3 | - | - | 2 | - | - | 6 | 12 | - |
| | 5,7 | 10 | - | - | - | 4 | - | - | 10 | - | 4 | - | 7 | - | 7 | - | 11 | - | 3 | - | 13 |
| | 5,8 | 10 | 7 | 11 | 8 | - | 10 | 8 | 9 | - | - | 9 | 12 | 13 | - | 11 | - | - | 3 | - | 5 |
| 6 | 6,1 | 15 | 15 | 9 | 6 | 9 | 14 | 13 | - | 8 | - | - | 5 | - | 3 | - | 15 | 8 | 7 | 15 | 14 |
| | 6,2 | - | - | - | - | 2 | 14 | 4 | - | - | - | 14 | 8 | - | - | 3 | - | - | 2 | - | - |
| | 6,3 | 12 | - | 9 | 10 | - | - | - | 4 | - | - | 10 | - | 4 | - | 7 | - | 7 | - | 11 | - |
| | 6,4 | - | - | 10 | 10 | 7 | 11 | 8 | - | 10 | 8 | 9 | - | - | 9 | 12 | 13 | - | 11 | - | - |
| | 6,5 | - | 7 | - | 15 | 15 | 9 | 6 | 9 | 14 | 13 | - | 8 | - | - | 5 | - | 3 | - | 15 | 8 |
| 7 | 7,1 | - | - | - | - | - | - | - | 2 | 14 | 4 | - | - | - | 14 | 8 | - | - | 3 | - | - |
| | 7,2 | 11 | - | 13 | 12 | - | 9 | 10 | - | - | - | 4 | - | - | 10 | - | 4 | - | 7 | - | 7 |
| | 7,3 | 2 | 12 | - | - | - | 10 | 10 | 7 | 11 | 8 | - | 10 | 8 | 9 | - | - | 9 | 12 | 13 | - |
| | 7,4 | - | - | 3 | - | 7 | - | 15 | 15 | 9 | 6 | 9 | 14 | 13 | - | 8 | - | - | 5 | - | 3 |
| | 7,5 | 2 | - | 11 | - | - | - | - | - | - | - | 2 | 14 | 4 | - | - | - | 14 | 8 | - | - |

*Table A.28 DR5 due dates and weights.*

| Job | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Due date $d_j$ | 15 | 11 | 20 | 19 | 13 | 13 | 13 |
| Weight $W_j$ | 6 | 8 | 3 | 2 | 1 | 4 | 9 |

*Table A.29 Due dates and weights FJSP-2F instance PP1.*

| Job | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Due date $d_j$ | 20 | 21 | 14 | 10 | 13 | 19 | 18 | 14 | 23 | 18 |
| Weight $W_j$ | 10 | 10 | 8 | 8 | 9 | 2 | 1 | 4 | 6 | 2 |

*Table A.30 PP1 FJSP-2F instance 10 x 5 with 146 operations.*

| Jobs | Op. | Machines | | | | |
|---|---|---|---|---|---|---|
| *j* | 2.36 | M1 | M2 | M3 | M4 | M5 |
| | 1,1 | - | - | - | - | 14 |
| | 2,1 | - | - | 15 | - | - |
| | 3,1 | - | - | 13 | - | - |
| | 4,1 | - | - | - | 15 | - |
| 1 | 5,1 | 15 | - | - | 9 | 10 |
| | 1,2 | 8 | 7 | - | - | 14 |
| | 2,2 | 13 | 8 | - | - | - |
| | 3,2 | - | 8 | 9 | 11 | - |
| | 4,2 | - | 12 | - | 10 | - |
| | 1,1 | 14 | 10 | - | - | - |
| | 2,1 | - | - | - | 7 | - |
| | 3,1 | - | - | - | 8 | 10 |
| | 4,1 | 10 | 13 | - | - | 9 |
| | 5,1 | - | 9 | - | 11 | - |
| | 6,1 | 5 | - | - | 15 | - |
| | 1,2 | - | - | 6 | 7 | 10 |
| | 2,2 | 12 | - | - | - | - |
| 2 | 3,2 | - | - | 12 | 5 | - |
| | 4,2 | 10 | 10 | 5 | 10 | 7 |
| | 5,2 | 8 | - | - | 10 | 15 |
| | 6,2 | - | 7 | 7 | 7 | - |
| | 1,3 | 10 | - | 6 | - | - |
| | 2,3 | 15 | 5 | 13 | 11 | 5 |
| | 3,3 | - | 10 | 14 | 15 | - |
| | 4,3 | 5 | 5 | - | - | - |
| | 5,3 | 10 | - | 5 | - | 9 |
| | 1,1 | - | - | - | 14 | - |
| | 2,1 | - | 15 | - | - | - |
| | 3,1 | - | 13 | - | - | - |
| | 4,1 | - | - | 15 | - | 15 |
| | 5,1 | - | - | 9 | 10 | 8 |
| | 6,1 | 7 | - | - | 14 | 13 |
| | 7,1 | 8 | - | - | - | - |
| 3 | 8,1 | 8 | 9 | 11 | - | - |
| | 1,2 | 12 | - | 10 | - | 14 |
| | 2,2 | 10 | - | - | - | - |
| | 3,2 | - | - | 7 | - | - |
| | 4,2 | - | - | 8 | 10 | 10 |
| | 5,2 | 13 | - | - | 9 | - |
| | 6,2 | 9 | - | 11 | - | 5 |
| | 7,2 | - | - | 15 | - | - |
| | 8,2 | - | 6 | 7 | 10 | 12 |
| | 1,1 | - | 12 | 5 | - | 10 |
| | 2,1 | 10 | 5 | 10 | 7 | 8 |
| | 3,1 | - | - | 10 | 15 | - |
| 4 | 4,1 | 7 | 7 | 7 | - | 10 |
| | 1,2 | - | 6 | - | - | 15 |
| | 2,2 | 5 | 13 | 11 | 5 | - |
| | 3,2 | 10 | 14 | 15 | - | 5 |
| | 4,2 | 5 | - | - | - | 10 |
| | 5,2 | - | 5 | - | 9 | - |
| | 1,3 | - | - | 14 | - | - |
| 4 | 2,3 | 15 | - | - | - | - |
| | 3,3 | 13 | - | - | - | - |
| | 4,3 | - | 15 | - | 15 | - |
| | 5,3 | - | 9 | 10 | 8 | 7 |
| | 1,1 | - | - | 14 | 13 | 8 |
| | 2,1 | - | - | - | - | 8 |
| | 3,1 | 9 | 11 | - | - | 12 |
| | 4,1 | - | 10 | - | 14 | 10 |
| | 5,1 | - | 7 | - | - | - |
| 5 | 6,1 | - | 8 | 10 | 10 | 13 |
| | 1,2 | - | - | 9 | - | - |
| | 2,2 | - | - | - | - | 9 |
| | 3,2 | - | 11 | - | 5 | - |
| | 4,2 | - | 15 | - | - | - |
| | 5,2 | 6 | 7 | 10 | 12 | - |
| | 1,1 | 12 | 5 | - | 10 | 10 |
| | 2,1 | 5 | 10 | 7 | 8 | - |
| | 3,1 | - | 10 | 15 | - | 7 |
| | 4,1 | 7 | 7 | - | 10 | - |
| | 5,1 | 6 | - | - | 15 | 5 |
| | 6,1 | 13 | 11 | 5 | - | 10 |
| | 1,2 | 14 | 15 | - | 5 | 5 |
| | 2,2 | - | - | - | 10 | - |
| | 3,2 | 5 | - | 9 | - | - |
| | 4,2 | - | 14 | - | - | 15 |
| 6 | 5,2 | - | - | - | - | 13 |
| | 6,2 | 15 | - | 15 | - | - |
| | 1,3 | 9 | 10 | 8 | 7 | - |
| | 2,3 | - | 14 | 13 | 8 | - |
| | 3,3 | - | - | - | 8 | 9 |
| | 4,3 | 11 | - | - | 12 | - |
| | 1,4 | 10 | - | 14 | 10 | - |
| | 2,4 | 7 | - | - | - | - |
| | 3,4 | 8 | 10 | 10 | 13 | - |
| | 4,4 | - | 9 | - | - | - |
| | 5,4 | - | - | - | 9 | - |
| | 1,1 | 11 | - | 5 | - | - |
| | 2,1 | 15 | - | - | - | 6 |
| | 3,1 | 7 | 10 | 12 | - | - |
| | 4,1 | - | - | - | - | 12 |
| 7 | 5,1 | 5 | - | 10 | 10 | 5 |
| | 6,1 | 10 | 7 | 8 | - | - |
| | 7,1 | 10 | 15 | - | 7 | 7 |
| | 8,1 | 7 | - | 10 | - | 6 |
| | 1,2 | - | - | 15 | 5 | 13 |
| | 2,2 | 11 | 5 | - | 10 | 14 |

*Table A.31 PP1 FJSP-2F instance 10 x 5 with 146 operations (continued).*

| Jobs | Op. | Machines | | | | |
|---|---|---|---|---|---|---|
| *j* | 2.36 | M1 | M2 | M3 | M4 | M5 |
| 7 | 3,2 | 15 | - | 5 | 5 | - |
| | 4,2 | - | - | 10 | - | 5 |
| | 5,2 | - | 9 | - | - | - |
| | 6,2 | 14 | - | - | 15 | - |
| | 7,2 | - | - | - | 13 | - |
| | 8,2 | - | - | - | - | 15 |
| 8 | 1,1 | - | 15 | - | - | 9 |
| | 2,1 | 10 | 8 | 7 | - | - |
| | 3,1 | 14 | 13 | 8 | - | - |
| | 4,1 | - | - | 8 | 9 | 11 |
| | 5,1 | - | - | 12 | - | 10 |
| | 1,2 | - | 14 | 10 | - | - |
| | 2,2 | - | - | - | - | 7 |
| | 3,2 | - | - | - | - | 8 |
| | 4,2 | 10 | 10 | 13 | - | - |
| 9 | 1,1 | 9 | - | - | - | - |
| | 2,1 | - | - | 9 | - | 11 |
| | 3,1 | - | 5 | - | - | 15 |
| | 4,1 | - | - | - | 6 | 7 |
| | 5,1 | 10 | 12 | - | - | - |
| | 6,1 | - | - | - | 12 | 5 |
| | 1,2 | - | 10 | 10 | 5 | 10 |
| | 2,2 | 7 | 8 | - | - | 10 |
| | 3,2 | 15 | - | 7 | 7 | 7 |
| | 4,2 | - | 10 | - | 6 | - |
| | 5,2 | - | 15 | 5 | 13 | 11 |
| | 6,2 | 5 | - | 10 | 14 | 15 |
| | 1,3 | - | 5 | 5 | - | - |
| | 2,3 | - | 10 | - | 5 | - |
| | 3,3 | 9 | - | - | - | 14 |
| | 4,3 | - | - | 15 | - | - |
| | 5,3 | - | - | 13 | - | - |
| 10 | 1,1 | - | - | - | 15 | - |
| | 2,1 | 15 | - | - | 9 | 10 |
| | 3,1 | 8 | 7 | - | - | 14 |
| | 4,1 | 13 | 8 | - | - | - |
| | 5,1 | - | 8 | 9 | 11 | - |
| | 6,1 | - | 12 | - | 10 | - |
| | 7,1 | 14 | 10 | - | - | - |
| | 8,1 | - | - | - | 7 | - |
| | 1,2 | - | - | - | 8 | 10 |
| | 2,2 | 10 | 13 | - | - | 9 |
| | 3,2 | - | 9 | - | 11 | - |
| | 4,2 | 5 | - | - | 15 | - |
| | 5,2 | - | - | 6 | 7 | 10 |
| | 6,2 | 12 | - | - | - | - |
| | 7,2 | - | - | 12 | 5 | - |
| | 8,2 | 10 | 10 | 5 | 10 | 7 |

*Table A.32 PP1 Costs FJSP-2F instance 10 x 5 with 146 operations.*

| Jobs | Op. | Machines | | | | |
|---|---|---|---|---|---|---|
| | 2.36 | M1 | M2 | M3 | M4 | M5 |
| 1 | 1,1 | 0 | 0 | 0 | 0 | 5 |
| | 2,1 | 0 | 0 | 3 | 0 | 0 |
| | 3,1 | 0 | 0 | 14 | 0 | 0 |
| | 4,1 | 0 | 0 | 0 | 17 | 0 |
| | 5,1 | 8 | 0 | 0 | 17 | 6 |
| | 1,2 | 12 | 15 | 0 | 0 | 1 |
| | 2,2 | 1 | 13 | 0 | 0 | 0 |
| | 3,2 | 0 | 1 | 18 | 1 | 0 |
| | 4,2 | 0 | 17 | 0 | 3 | 0 |
| 2 | 1,1 | 16 | 2 | 0 | 0 | 0 |
| | 2,1 | 0 | 0 | 0 | 9 | 0 |
| | 3,1 | 0 | 0 | 0 | 2 | 13 |
| | 4,1 | 15 | 8 | 0 | 0 | 10 |
| | 5,1 | 0 | 4 | 0 | 13 | 0 |
| | 6,1 | 5 | 0 | 0 | 9 | 0 |
| | 1,2 | 0 | 0 | 7 | 18 | 14 |
| | 2,2 | 12 | 0 | 0 | 0 | 0 |
| | 3,2 | 0 | 0 | 10 | 2 | 0 |
| | 4,2 | 7 | 10 | 9 | 11 | 5 |
| | 5,2 | 6 | 0 | 0 | 17 | 5 |
| | 6,2 | 0 | 16 | 4 | 11 | 0 |
| | 1,3 | 1 | 0 | 12 | 0 | 0 |
| | 2,3 | 10 | 12 | 3 | 6 | 1 |
| | 3,3 | 0 | 8 | 12 | 5 | 0 |
| | 4,3 | 14 | 14 | 0 | 0 | 0 |
| | 5,3 | 5 | 0 | 3 | 0 | 6 |
| 3 | 1,1 | 0 | 0 | 0 | 5 | 0 |
| | 2,1 | 0 | 3 | 0 | 0 | 0 |
| | 3,1 | 0 | 14 | 0 | 0 | 0 |
| | 4,1 | 0 | 0 | 17 | 0 | 8 |
| | 5,1 | 0 | 0 | 17 | 6 | 12 |
| | 6,1 | 15 | 0 | 0 | 1 | 1 |
| | 7,1 | 13 | 0 | 0 | 0 | 0 |
| | 8,1 | 1 | 18 | 1 | 0 | 0 |
| | 1,2 | 17 | 0 | 3 | 0 | 16 |
| | 2,2 | 2 | 0 | 0 | 0 | 0 |
| | 3,2 | 0 | 0 | 9 | 0 | 0 |
| | 4,2 | 0 | 0 | 2 | 13 | 15 |
| | 5,2 | 8 | 0 | 0 | 10 | 0 |
| | 6,2 | 4 | 0 | 13 | 0 | 5 |
| | 7,2 | 0 | 0 | 9 | 0 | 0 |
| | 8,2 | 0 | 7 | 18 | 14 | 12 |
| 4 | 1,1 | 0 | 10 | 2 | 0 | 7 |
| | 2,1 | 10 | 9 | 11 | 5 | 6 |
| | 3,1 | 0 | 0 | 17 | 5 | 0 |
| | 4,1 | 16 | 4 | 11 | 0 | 1 |
| | 1,2 | 0 | 12 | 0 | 0 | 10 |
| | 2,2 | 12 | 3 | 6 | 1 | 0 |
| | 3,2 | 8 | 12 | 5 | 0 | 14 |
| | 4,2 | 14 | 0 | 0 | 0 | 5 |
| | 5,2 | 0 | 3 | 0 | 6 | 0 |
| | 1,3 | 0 | 0 | 5 | 0 | 0 |
| | 2,3 | 3 | 0 | 0 | 0 | 0 |
| | 3,3 | 14 | 0 | 0 | 0 | 0 |
| | 4,3 | 0 | 17 | 0 | 8 | 0 |
| | 5,3 | 0 | 17 | 6 | 12 | 15 |
| 5 | 1,1 | 0 | 0 | 1 | 1 | 13 |
| | 2,1 | 0 | 0 | 0 | 0 | 1 |
| | 3,1 | 18 | 1 | 0 | 0 | 17 |
| | 4,1 | 0 | 3 | 0 | 16 | 2 |
| | 5,1 | 0 | 9 | 0 | 0 | 0 |
| | 6,1 | 0 | 2 | 13 | 15 | 8 |
| | 1,2 | 0 | 0 | 10 | 0 | 0 |
| | 2,2 | 0 | 0 | 0 | 0 | 4 |
| | 3,2 | 0 | 13 | 0 | 5 | 0 |
| | 4,2 | 0 | 9 | 0 | 0 | 0 |
| | 5,2 | 7 | 18 | 14 | 12 | 0 |
| 6 | 1,1 | 10 | 2 | 0 | 7 | 10 |
| | 2,1 | 9 | 11 | 5 | 6 | 0 |
| | 3,1 | 0 | 17 | 5 | 0 | 16 |
| | 4,1 | 4 | 11 | 0 | 1 | 0 |
| | 5,1 | 12 | 0 | 0 | 10 | 12 |
| | 6,1 | 3 | 6 | 1 | 0 | 8 |
| | 1,2 | 12 | 5 | 0 | 14 | 14 |
| | 2,2 | 0 | 0 | 0 | 5 | 0 |
| | 3,2 | 3 | 0 | 6 | 0 | 0 |
| | 4,2 | 0 | 5 | 0 | 0 | 3 |
| | 5,2 | 0 | 0 | 0 | 0 | 14 |
| | 6,2 | 17 | 0 | 8 | 0 | 0 |
| | 1,3 | 17 | 6 | 12 | 15 | 0 |
| | 2,3 | 0 | 1 | 1 | 13 | 0 |
| | 3,3 | 0 | 0 | 0 | 1 | 18 |
| | 4,3 | 1 | 0 | 0 | 17 | 0 |
| | 1,4 | 3 | 0 | 16 | 2 | 0 |
| | 2,4 | 9 | 0 | 0 | 0 | 0 |
| | 3,4 | 2 | 13 | 15 | 8 | 0 |
| | 4,4 | 0 | 10 | 0 | 0 | 0 |
| | 5,4 | 0 | 0 | 0 | 4 | 0 |
| 7 | 1,1 | 13 | 0 | 5 | 0 | 0 |
| | 2,1 | 9 | 0 | 0 | 0 | 7 |
| | 3,1 | 18 | 14 | 12 | 0 | 0 |
| | 4,1 | 0 | 0 | 0 | 0 | 10 |
| | 5,1 | 2 | 0 | 7 | 10 | 9 |
| | 6,1 | 11 | 5 | 6 | 0 | 0 |
| | 7,1 | 17 | 5 | 0 | 16 | 4 |
| | 8,1 | 11 | 0 | 1 | 0 | 12 |
| | 1,2 | 0 | 0 | 10 | 12 | 3 |
| | 2,2 | 6 | 1 | 0 | 8 | 12 |

Table A.33 PP1 Costs FJSP-2F instance 10 x 5 with 146 operations (continued).

| Jobs | Op. | Machines | | | | |
|---|---|---|---|---|---|---|
| | 2.36 | M1 | M2 | M3 | M4 | M5 |
| 7 | 3,2 | 5 | 0 | 14 | 14 | 0 |
| | 4,2 | 0 | 0 | 5 | 0 | 3 |
| | 5,2 | 0 | 6 | 0 | 0 | 0 |
| | 6,2 | 5 | 0 | 0 | 3 | 0 |
| | 7,2 | 0 | 0 | 0 | 14 | 0 |
| | 8,2 | 0 | 0 | 0 | 0 | 17 |
| 8 | 1,1 | 0 | 8 | 0 | 0 | 17 |
| | 2,1 | 6 | 12 | 15 | 0 | 0 |
| | 3,1 | 1 | 1 | 13 | 0 | 0 |
| | 4,1 | 0 | 0 | 1 | 18 | 1 |
| | 5,1 | 0 | 0 | 17 | 0 | 3 |
| | 1,2 | 0 | 16 | 2 | 0 | 0 |
| | 2,2 | 0 | 0 | 0 | 0 | 9 |
| | 3,2 | 0 | 0 | 0 | 0 | 2 |
| | 4,2 | 13 | 15 | 8 | 0 | 0 |
| 9 | 1,1 | 10 | 0 | 0 | 0 | 0 |
| | 2,1 | 0 | 0 | 4 | 0 | 13 |
| | 3,1 | 0 | 5 | 0 | 0 | 9 |
| | 4,1 | 0 | 0 | 0 | 7 | 18 |
| | 5,1 | 14 | 12 | 0 | 0 | 0 |
| | 6,1 | 0 | 0 | 0 | 10 | 2 |
| | 1,2 | 0 | 7 | 10 | 9 | 11 |
| | 2,2 | 5 | 6 | 0 | 0 | 17 |
| | 3,2 | 5 | 0 | 16 | 4 | 11 |
| | 4,2 | 0 | 1 | 0 | 12 | 0 |
| | 5,2 | 0 | 10 | 12 | 3 | 6 |
| | 6,2 | 1 | 0 | 8 | 12 | 5 |
| | 1,3 | 0 | 14 | 14 | 0 | 0 |
| | 2,3 | 0 | 5 | 0 | 3 | 0 |
| | 3,3 | 6 | 0 | 0 | 0 | 5 |
| | 4,3 | 0 | 0 | 3 | 0 | 0 |
| | 5,3 | 0 | 0 | 14 | 0 | 0 |
| 10 | 1,1 | 0 | 0 | 0 | 17 | 0 |
| | 2,1 | 8 | 0 | 0 | 17 | 6 |
| | 3,1 | 12 | 15 | 0 | 0 | 1 |
| | 4,1 | 1 | 13 | 0 | 0 | 0 |
| | 5,1 | 0 | 1 | 18 | 1 | 0 |
| | 6,1 | 0 | 17 | 0 | 3 | 0 |
| | 7,1 | 16 | 2 | 0 | 0 | 0 |
| | 8,1 | 0 | 0 | 0 | 9 | 0 |
| | 1,2 | 0 | 0 | 0 | 2 | 13 |
| | 2,2 | 15 | 8 | 0 | 0 | 10 |
| | 3,2 | 0 | 4 | 0 | 13 | 0 |
| | 4,2 | 5 | 0 | 0 | 9 | 0 |
| | 5,2 | 0 | 0 | 7 | 18 | 14 |
| | 6,2 | 12 | 0 | 0 | 0 | 0 |
| | 7,2 | 0 | 0 | 10 | 2 | 0 |
| | 8,2 | 7 | 10 | 9 | 11 | 5 |

*Table A.34 PP2 FJSP-2F instance 5 x 8 with 70 operations.*

| Jobs | Op. | Machines | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 3.94 | k,p | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 |
| 1 | 1,1 | - | - | - | - | - | - | - | 3 |
| | 2,1 | 7 | 9 | 2 | 5 | - | - | - | - |
| | 3,1 | 3 | 8 | 4 | 9 | 6 | - | 10 | 4 |
| | 4,1 | - | - | - | - | - | - | 2 | - |
| | 5,1 | - | - | - | 2 | 4 | 4 | 9 | - |
| | 6,1 | - | 6 | 6 | - | - | - | 4 | - |
| | 7,1 | 7 | - | 9 | 4 | 8 | - | - | 9 |
| | 1,2 | 2 | 7 | 8 | - | - | 2 | - | 8 |
| | 2,2 | 2 | - | 7 | - | 8 | 6 | - | - |
| | 3,2 | 10 | 3 | - | 2 | 7 | 3 | - | - |
| | 4,2 | - | - | 9 | - | - | 7 | - | 10 |
| | 5,2 | - | - | 9 | 9 | 9 | 7 | 9 | - |
| | 6,2 | 4 | 8 | 8 | - | 9 | 3 | 2 | - |
| | 7,2 | 9 | 9 | - | 5 | - | - | - | 10 |
| | 8,2 | - | - | 2 | - | 3 | 9 | - | - |
| | 1,3 | - | - | - | 8 | - | 8 | - | - |
| | 2,3 | 6 | 4 | - | - | - | - | - | 3 |
| | 3,3 | 7 | 9 | 2 | 5 | - | - | - | - |
| | 4,3 | 3 | 8 | 4 | 9 | 6 | - | 10 | 4 |
| | 5,3 | - | - | - | - | - | - | 2 | - |
| | 6,3 | - | - | - | 2 | 4 | 4 | 9 | - |
| | 7,3 | - | 6 | 6 | - | - | - | 4 | - |
| | 8,3 | 7 | - | 9 | 4 | 8 | - | - | 9 |
| 2 | 1,1 | 2 | 7 | 8 | - | - | 2 | - | 8 |
| | 2,1 | 2 | - | 7 | - | 8 | 6 | - | - |
| | 3,1 | 10 | 3 | - | 2 | 7 | 3 | - | - |
| | 4,1 | - | - | 9 | - | - | 7 | - | 10 |
| | 5,1 | - | - | 9 | 9 | 9 | 7 | 9 | - |
| | 1,2 | 4 | 8 | 8 | - | 9 | 3 | 2 | - |
| | 2,2 | 9 | 9 | - | 5 | - | - | - | 10 |
| | 3,2 | - | - | 2 | - | 3 | 9 | - | - |
| | 4,2 | - | - | - | 8 | - | 8 | - | - |
| | 5,2 | 6 | 4 | - | - | - | - | - | 3 |
| 3 | 1,1 | 7 | 9 | 2 | 5 | - | - | - | - |
| | 2,1 | 3 | 8 | 4 | 9 | 6 | - | 10 | 4 |
| | 3,1 | - | - | - | - | - | - | 2 | - |
| | 4,1 | - | - | - | 2 | 4 | 4 | 9 | - |
| | 5,1 | - | 6 | 6 | - | - | - | 4 | - |
| | 6,1 | 7 | - | 9 | 4 | 8 | - | - | 9 |
| | 1,2 | 2 | 7 | 8 | - | - | 2 | - | 8 |
| | 2,2 | 2 | - | 7 | - | 8 | 6 | - | - |
| | 3,2 | 10 | 3 | - | 2 | 7 | 3 | - | - |
| | 4,2 | - | - | 9 | - | - | 7 | - | 10 |
| | 5,2 | - | - | 9 | 9 | 9 | 7 | 9 | - |
| | 6,2 | 4 | 8 | 8 | - | 9 | 3 | 2 | - |

*Table A.35 PP2 FJSP-2F instance 5 x 8 with 70 operations (continued).*

| Jobs | Op. | Machines | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 3.94 | k,p | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 |
| 4 | 1,1 | 9 | 9 | - | 5 | - | - | - | 10 |
| | 2,1 | - | - | 2 | - | 3 | 9 | - | - |
| | 3,1 | - | - | - | 8 | - | 8 | - | - |
| | 4,1 | 6 | 4 | - | - | - | - | - | 3 |
| | 5,1 | 7 | 9 | 2 | 5 | - | - | - | - |
| | 1,2 | 3 | 8 | 4 | 9 | 6 | - | 10 | 4 |
| | 2,2 | - | - | - | - | - | - | 2 | - |
| | 3,2 | - | - | - | 2 | 4 | 4 | 9 | - |
| | 4,2 | - | 6 | 6 | - | - | - | 4 | - |
| | 5,2 | 7 | - | 9 | 4 | 8 | - | - | 9 |
| | 1,3 | 2 | 7 | 8 | - | - | 2 | - | 8 |
| | 2,3 | 2 | - | 7 | - | 8 | 6 | - | - |
| | 3,3 | 10 | 3 | - | 2 | 7 | 3 | - | - |
| | 4,3 | - | - | 9 | - | - | 7 | - | 10 |
| 5 | 1,1 | - | - | 9 | 9 | 9 | 7 | 9 | - |
| | 2,1 | 4 | 8 | 8 | - | 9 | 3 | 2 | - |
| | 3,1 | 9 | 9 | - | 5 | - | - | - | 10 |
| | 4,1 | - | - | 2 | - | 3 | 9 | - | - |
| | 5,1 | - | - | - | 8 | - | 8 | - | - |
| | 6,1 | 6 | 4 | - | - | - | - | - | 3 |
| | 1,2 | 7 | 9 | 2 | 5 | - | - | - | - |
| | 2,2 | 3 | 8 | 4 | 9 | 6 | - | 10 | 4 |
| | 3,2 | - | - | - | - | - | - | 2 | - |
| | 4,2 | - | - | - | 2 | 4 | 4 | 9 | - |
| | 5,2 | - | 6 | 6 | - | - | - | 4 | - |

*Table A.36 Due dates and weights FJSP-2F instance PP2.*

| Job | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Due date $d_j$ | 13 | 11 | 16 | 18 | 12 |
| Weight $W_j$ | 6 | 10 | 9 | 6 | 2 |

*Table A.37 Costs PP2 FJSP-2F instance 5 x 8 with 70 operations.*

| Jobs | Op. | Machines | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 3.94 | k,p | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 |
| 1 | 1,1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 |
| | 2,1 | 12 | 6 | 9 | 11 | 0 | 0 | 0 | 0 |
| | 3,1 | 11 | 6 | 10 | 14 | 7 | 0 | 15 | 13 |
| | 4,1 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 0 |
| | 5,1 | 0 | 0 | 0 | 13 | 9 | 10 | 10 | 0 |
| | 6,1 | 0 | 11 | 7 | 0 | 0 | 0 | 14 | 0 |
| | 7,1 | 13 | 0 | 15 | 6 | 12 | 0 | 0 | 5 |
| | 1,2 | 12 | 14 | 13 | 0 | 0 | 5 | 0 | 11 |
| | 2,2 | 7 | 0 | 10 | 0 | 7 | 6 | 0 | 0 |
| | 3,2 | 15 | 5 | 0 | 14 | 10 | 7 | 0 | 0 |
| | 4,2 | 0 | 0 | 12 | 0 | 0 | 6 | 0 | 8 |
| | 5,2 | 0 | 0 | 7 | 8 | 7 | 9 | 12 | 0 |
| | 6,2 | 6 | 8 | 15 | 0 | 9 | 13 | 9 | 0 |
| | 7,2 | 15 | 13 | 0 | 9 | 0 | 0 | 0 | 8 |
| | 8,2 | 0 | 0 | 7 | 0 | 12 | 14 | 0 | 0 |
| | 1,3 | 0 | 0 | 0 | 11 | 0 | 5 | 0 | 0 |
| | 2,3 | 15 | 15 | 0 | 0 | 0 | 0 | 0 | 8 |
| | 3,3 | 12 | 6 | 9 | 11 | 0 | 0 | 0 | 0 |
| | 4,3 | 11 | 6 | 10 | 14 | 7 | 0 | 15 | 13 |
| | 5,3 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 0 |
| | 6,3 | 0 | 0 | 0 | 13 | 9 | 10 | 10 | 0 |
| | 7,3 | 0 | 11 | 7 | 0 | 0 | 0 | 14 | 0 |
| | 8,3 | 13 | 0 | 15 | 6 | 12 | 0 | 0 | 5 |
| 2 | 1,1 | 12 | 14 | 13 | 0 | 0 | 5 | 0 | 11 |
| | 2,1 | 7 | 0 | 10 | 0 | 7 | 6 | 0 | 0 |
| | 3,1 | 15 | 5 | 0 | 14 | 10 | 7 | 0 | 0 |
| | 4,1 | 0 | 0 | 12 | 0 | 0 | 6 | 0 | 8 |
| | 5,1 | 0 | 0 | 7 | 8 | 7 | 9 | 12 | 0 |
| | 1,2 | 6 | 8 | 15 | 0 | 9 | 13 | 9 | 0 |
| | 2,2 | 15 | 13 | 0 | 9 | 0 | 0 | 0 | 8 |
| | 3,2 | 0 | 0 | 7 | 0 | 12 | 14 | 0 | 0 |
| | 4,2 | 0 | 0 | 0 | 11 | 0 | 5 | 0 | 0 |
| | 5,2 | 15 | 15 | 0 | 0 | 0 | 0 | 0 | 8 |
| 3 | 1,1 | 12 | 6 | 9 | 11 | 0 | 0 | 0 | 0 |
| | 2,1 | 11 | 6 | 10 | 14 | 7 | 0 | 15 | 13 |
| | 3,1 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 0 |
| | 4,1 | 0 | 0 | 0 | 13 | 9 | 10 | 10 | 0 |
| | 5,1 | 0 | 11 | 7 | 0 | 0 | 0 | 14 | 0 |
| | 6,1 | 13 | 0 | 15 | 6 | 12 | 0 | 0 | 5 |
| | 1,2 | 12 | 14 | 13 | 0 | 0 | 5 | 0 | 11 |
| | 2,2 | 7 | 0 | 10 | 0 | 7 | 6 | 0 | 0 |
| | 3,2 | 15 | 5 | 0 | 14 | 10 | 7 | 0 | 0 |
| | 4,2 | 0 | 0 | 12 | 0 | 0 | 6 | 0 | 8 |
| | 5,2 | 0 | 0 | 7 | 8 | 7 | 9 | 12 | 0 |
| | 6,2 | 6 | 8 | 15 | 0 | 9 | 13 | 9 | 0 |

*Table A.38 Costs PP2 FJSP-2F instance 5 x 8 with 70 operations (continued).*

| Jobs | Op. | Machines | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 3.94 | k,p | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 |
| 4 | 1,1 | 15 | 13 | 0 | 9 | 0 | 0 | 0 | 8 |
| | 2,1 | 0 | 0 | 7 | 0 | 12 | 14 | 0 | 0 |
| | 3,1 | 0 | 0 | 0 | 11 | 0 | 5 | 0 | 0 |
| | 4,1 | 15 | 15 | 0 | 0 | 0 | 0 | 0 | 8 |
| | 5,1 | 12 | 6 | 9 | 11 | 0 | 0 | 0 | 0 |
| | 1,2 | 11 | 6 | 10 | 14 | 7 | 0 | 15 | 13 |
| | 2,2 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 0 |
| | 3,2 | 0 | 0 | 0 | 13 | 9 | 10 | 10 | 0 |
| | 4,2 | 0 | 11 | 7 | 0 | 0 | 0 | 14 | 0 |
| | 5,2 | 13 | 0 | 15 | 6 | 12 | 0 | 0 | 5 |
| | 1,3 | 12 | 14 | 13 | 0 | 0 | 5 | 0 | 11 |
| | 2,3 | 7 | 0 | 10 | 0 | 7 | 6 | 0 | 0 |
| | 3,3 | 15 | 5 | 0 | 14 | 10 | 7 | 0 | 0 |
| | 4,3 | 0 | 0 | 12 | 0 | 0 | 6 | 0 | 8 |
| 5 | 1,1 | 0 | 0 | 7 | 8 | 7 | 9 | 12 | 0 |
| | 2,1 | 6 | 8 | 15 | 0 | 9 | 13 | 9 | 0 |
| | 3,1 | 15 | 13 | 0 | 9 | 0 | 0 | 0 | 8 |
| | 4,1 | 0 | 0 | 7 | 0 | 12 | 14 | 0 | 0 |
| | 5,1 | 0 | 0 | 0 | 11 | 0 | 5 | 0 | 0 |
| | 6,1 | 15 | 15 | 0 | 0 | 0 | 0 | 0 | 8 |
| | 1,2 | 12 | 6 | 9 | 11 | 0 | 0 | 0 | 0 |
| | 2,2 | 11 | 6 | 10 | 14 | 7 | 0 | 15 | 13 |
| | 3,2 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 0 |
| | 4,2 | 0 | 0 | 0 | 13 | 9 | 10 | 10 | 0 |
| | 5,2 | 0 | 11 | 7 | 0 | 0 | 0 | 14 | 0 |

*Table A.39 PP3 FJSP-2F instance 6 x 6 with 111 operations.*

| Jobs | Op. | Machines | | | | | |
|---|---|---|---|---|---|---|---|
| 3.63 | k,p | M1 | M2 | M3 | M4 | M5 | M6 |
| 1 | 1,1 | 6 | - | 6 | - | 7 | 5 |
| | 2,1 | - | - | 8 | - | 4 | - |
| | 3,1 | - | - | - | 4 | 5 | 5 |
| | 4,1 | 6 | 4 | 7 | - | 6 | - |
| | 5,1 | 5 | - | 5 | 7 | 8 | 7 |
| | 6,1 | 6 | 8 | 4 | 6 | 6 | - |
| | 1,2 | - | 8 | - | 6 | - | 6 |
| | 2,2 | 7 | 7 | 8 | 5 | 4 | 4 |
| | 3,2 | - | 8 | 6 | 4 | - | - |
| | 4,2 | - | - | 7 | - | - | 5 |
| | 5,2 | - | 8 | 5 | 8 | 7 | 4 |
| | 6,2 | - | 8 | 4 | - | - | 7 |
| | 7,2 | 7 | - | 8 | - | 5 | 6 |
| | 8,2 | 7 | 8 | - | - | - | 5 |

*Table A.40 PP3 FJSP-2F instance 6  x 6 with 111 operations (continued).*

| Jobs | Op. | Machines | | | | | |
|---|---|---|---|---|---|---|---|
| 3.63 | k,p | M1 | M2 | M3 | M4 | M5 | M6 |
| | 1,1 | - | 7 | - | - | 7 | - |
| | 2,1 | 8 | 6 | 4 | - | - | 7 |
| | 3,1 | 7 | - | - | - | - | - |
| | 4,1 | 7 | 4 | 4 | 8 | 7 | 7 |
| | 5,1 | 4 | 6 | - | - | - | - |
| | 6,1 | 5 | 6 | 5 | - | 4 | 6 |
| | 1,2 | 7 | 5 | - | - | 8 | - |
| | 2,2 | 4 | - | - | - | - | 4 |
| | 3,2 | 5 | 5 | 6 | 4 | 7 | - |
| | 4,2 | 6 | - | 5 | - | 5 | 7 |
| | 5,2 | 8 | 7 | 6 | 8 | 4 | 6 |
| 2 | 1,3 | 6 | - | - | 8 | - | 6 |
| | 2,3 | - | 6 | 7 | 7 | 8 | 5 |
| | 3,3 | 4 | 4 | - | 8 | 6 | 4 |
| | 4,3 | - | - | - | - | 7 | - |
| | 5,3 | - | 5 | - | 8 | 5 | 8 |
| | 6,3 | 7 | 4 | - | 8 | 4 | - |
| | 7,3 | - | 7 | 7 | - | 8 | - |
| | 8,3 | 5 | 6 | 7 | 8 | - | - |
| | 1,4 | - | 5 | - | 7 | - | - |
| | 2,4 | 7 | - | 8 | 6 | 4 | - |
| | 3,4 | - | 7 | 7 | - | - | - |
| | 4,4 | - | - | 7 | 4 | 4 | 8 |
| | 5,4 | 7 | 7 | 4 | 6 | - | - |
| | 1,1 | - | - | 5 | 6 | 5 | - |
| | 2,1 | 4 | 6 | 7 | 5 | - | - |
| | 3,1 | 8 | - | 4 | - | - | - |
| | 4,1 | - | 4 | 5 | 5 | 6 | 4 |
| | 5,1 | 7 | - | 6 | - | 5 | - |
| | 6,1 | 5 | 7 | 8 | 7 | 6 | 8 |
| | 7,1 | 4 | 6 | 6 | - | - | 8 |
| | 8,1 | - | 6 | - | 6 | 7 | 7 |
| | 1,2 | 8 | 5 | 4 | 4 | - | 8 |
| | 2,2 | 6 | 4 | - | - | - | - |
| 3 | 3,2 | 7 | - | - | 5 | - | 8 |
| | 4,2 | 5 | 8 | 7 | 4 | - | 8 |
| | 5,2 | 4 | - | - | 7 | 7 | - |
| | 6,2 | 8 | - | 5 | 6 | 7 | 8 |
| | 1,3 | - | - | - | 5 | - | 7 |
| | 2,3 | - | - | 7 | - | 8 | 6 |
| | 3,3 | 4 | - | - | 7 | 7 | - |
| | 4,3 | - | - | - | - | 7 | 4 |
| | 5,3 | 4 | 8 | 7 | 7 | 4 | 6 |
| | 6,3 | - | - | - | - | 5 | 6 |
| | 7,3 | 5 | - | 4 | 6 | 7 | 5 |

| Jobs | Op. | Machines | | | | | |
|---|---|---|---|---|---|---|---|
| 3.943 | k,p | M1 | M2 | M3 | M4 | M5 | M6 |
| | 1,1 | - | - | 8 | - | 4 | - |
| | 2,1 | - | - | - | 4 | 5 | 5 |
| | 3,1 | 6 | 4 | 7 | - | 6 | - |
| | 4,1 | 5 | - | 5 | 7 | 8 | 7 |
| | 5,1 | 6 | 8 | 4 | 6 | 6 | - |
| | 6,1 | - | 8 | - | 6 | - | 6 |
| | 7,1 | 7 | 7 | 8 | 5 | 4 | 4 |
| 4 | 8,1 | - | 8 | 6 | 4 | - | - |
| | 1,2 | - | - | 7 | - | - | 5 |
| | 2,2 | - | 8 | 5 | 8 | 7 | 4 |
| | 3,2 | - | 8 | 4 | - | - | 7 |
| | 4,2 | 7 | - | 8 | - | 5 | 6 |
| | 5,2 | 7 | 8 | - | - | - | 5 |
| | 6,2 | - | 7 | - | - | 7 | - |
| | 7,2 | 8 | 6 | 4 | - | - | 7 |
| | 8,2 | 7 | - | - | - | - | - |
| | 1,1 | 7 | 4 | 4 | 8 | 7 | 7 |
| | 2,1 | 4 | 6 | - | - | - | - |
| | 3,1 | 5 | 6 | 5 | - | 4 | 6 |
| | 4,1 | 7 | 5 | - | - | 8 | - |
| | 5,1 | 4 | - | - | - | - | 4 |
| | 6,1 | 5 | 5 | 6 | 4 | 7 | - |
| | 1,2 | 6 | - | 5 | - | 5 | 7 |
| | 2,2 | 8 | 7 | 6 | 8 | 4 | 6 |
| | 3,2 | 6 | - | - | 8 | - | 6 |
| | 4,2 | - | 6 | 7 | 7 | 8 | 5 |
| 5 | 1,3 | 4 | 4 | - | 8 | 6 | 4 |
| | 2,3 | - | - | - | - | 7 | - |
| | 3,3 | - | 5 | - | 8 | 5 | 8 |
| | 4,3 | 7 | 4 | - | 8 | 4 | - |
| | 5,3 | - | 7 | 7 | - | 8 | - |
| | 6,3 | 5 | 6 | 7 | 8 | - | - |
| | 1,4 | - | 5 | - | 7 | - | - |
| | 2,4 | 7 | - | 8 | 6 | 4 | - |
| | 3,4 | - | 7 | 7 | - | - | - |
| | 4,4 | - | - | 7 | 4 | 4 | 8 |
| | 5,4 | 7 | 7 | 4 | 6 | - | - |
| | 1,1 | - | - | 5 | 6 | 5 | - |
| | 2,1 | 4 | 6 | 7 | 5 | - | - |
| | 3,1 | 8 | - | 4 | - | - | - |
| 6 | 4,1 | - | 4 | 5 | 5 | 6 | 4 |
| | 5,1 | 7 | - | 6 | - | 5 | - |
| | 6,1 | 5 | 7 | 8 | 7 | 6 | 8 |
| | 7,1 | 4 | 6 | 6 | - | - | 8 |
| | 1,2 | - | 6 | - | 6 | 7 | 7 |

*Table A.41 PP3 FJSP-2F instance 6 x 6 with 111 operations (continued).*

| Jobs | Op. | Machines | | | | | |
|---|---|---|---|---|---|---|---|
| 3.63 | k,p | M1 | M2 | M3 | M4 | M5 | M6 |
| | 2,2 | 8 | 5 | 4 | 4 | - | 8 |
| | 3,2 | 6 | 4 | - | - | - | - |
| | 4,2 | 7 | - | - | 5 | - | 8 |
| 6 | 5,2 | 5 | 8 | 7 | 4 | - | 8 |
| | 6,2 | 4 | - | - | 7 | 7 | - |
| | 7,2 | 8 | - | 5 | 6 | 7 | 8 |
| | 8,2 | - | - | - | 5 | - | 7 |

*Table A.42 Due dates and weights PP3 FJSP-2F instance.*

| Job | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Due date $d_j$ | 15 | 19 | 13 | 10 | 18 | 14 |
| Weight $W_j$ | 5 | 7 | 8 | 7 | 4 | 7 |

*Table A.43 Costs PP3 FJSP-2F instance 6 x 6 with 111 operations.*

| Jobs | Op. | Machines | | | | | |
|---|---|---|---|---|---|---|---|
| 3.63 | k,p | M1 | M2 | M3 | M4 | M5 | M6 |
| | 1,1 | 19 | 0 | 15 | 0 | 18 | 20 |
| | 2,1 | 0 | 0 | 18 | 0 | 14 | 0 |
| | 3,1 | 0 | 0 | 0 | 20 | 16 | 17 |
| | 4,1 | 13 | 14 | 10 | 0 | 17 | 0 |
| | 5,1 | 11 | 0 | 16 | 17 | 14 | 14 |
| | 6,1 | 12 | 13 | 19 | 10 | 16 | 0 |
| 1 | 1,2 | 0 | 19 | 0 | 14 | 0 | 11 |
| | 2,2 | 17 | 19 | 14 | 15 | 11 | 12 |
| | 3,2 | 0 | 17 | 13 | 15 | 0 | 0 |
| | 4,2 | 0 | 0 | 15 | 0 | 0 | 20 |
| | 5,2 | 0 | 15 | 14 | 13 | 20 | 20 |
| | 6,2 | 0 | 15 | 11 | 0 | 0 | 16 |
| | 7,2 | 11 | 0 | 17 | 0 | 12 | 13 |
| | 8,2 | 18 | 10 | 0 | 0 | 0 | 19 |

*Table A.44 Costs PP3 FJSP-2F instance 6 x 6 with 111 operations (continued).*

| Jobs | Op. | Machines | | | | | |
|---|---|---|---|---|---|---|---|
| 3.63 | k,p | M1 | M2 | M3 | M4 | M5 | M6 |
| | 1,1 | 0 | 11 | 0 | 0 | 10 | 0 |
| | 2,1 | 15 | 17 | 15 | 0 | 0 | 18 |
| | 3,1 | 13 | 0 | 0 | 0 | 0 | 0 |
| | 4,1 | 14 | 17 | 13 | 11 | 20 | 20 |
| | 5,1 | 16 | 19 | 0 | 0 | 0 | 0 |
| | 6,1 | 19 | 20 | 18 | 0 | 11 | 11 |
| | 1,2 | 18 | 20 | 0 | 0 | 18 | 0 |
| | 2,2 | 14 | 0 | 0 | 0 | 0 | 20 |
| | 3,2 | 16 | 17 | 13 | 14 | 10 | 0 |
| | 4,2 | 17 | 0 | 11 | 0 | 16 | 17 |
| | 5,2 | 14 | 14 | 12 | 13 | 19 | 10 |
| 2 | 1,3 | 16 | 0 | 0 | 19 | 0 | 14 |
| | 2,3 | 0 | 11 | 17 | 19 | 14 | 15 |
| | 3,3 | 11 | 12 | 0 | 17 | 13 | 15 |
| | 4,3 | 0 | 0 | 0 | 0 | 15 | 0 |
| | 5,3 | 0 | 20 | 0 | 15 | 14 | 13 |
| | 6,3 | 20 | 20 | 0 | 15 | 11 | 0 |
| | 7,3 | 0 | 16 | 11 | 0 | 17 | 0 |
| | 8,3 | 12 | 13 | 18 | 10 | 0 | 0 |
| | 1,4 | 0 | 19 | 0 | 11 | 0 | 0 |
| | 2,4 | 10 | 0 | 15 | 17 | 15 | 0 |
| | 3,4 | 0 | 18 | 13 | 0 | 0 | 0 |
| | 4,4 | 0 | 0 | 14 | 17 | 13 | 11 |
| | 5,4 | 20 | 20 | 16 | 19 | 0 | 0 |
| | 1,1 | 0 | 0 | 19 | 20 | 18 | 0 |
| | 2,1 | 11 | 11 | 18 | 20 | 0 | 0 |
| | 3,1 | 18 | 0 | 14 | 0 | 0 | 0 |
| | 4,1 | 0 | 20 | 16 | 17 | 13 | 14 |
| | 5,1 | 10 | 0 | 17 | 0 | 11 | 0 |
| | 6,1 | 16 | 17 | 14 | 14 | 12 | 13 |
| | 7,1 | 19 | 10 | 16 | 0 | 0 | 19 |
| | 8,1 | 0 | 14 | 0 | 11 | 17 | 19 |
| | 1,2 | 14 | 15 | 11 | 12 | 0 | 17 |
| | 2,2 | 13 | 15 | 0 | 0 | 0 | 0 |
| 3 | 3,2 | 15 | 0 | 0 | 20 | 0 | 15 |
| | 4,2 | 14 | 13 | 20 | 20 | 0 | 15 |
| | 5,2 | 11 | 0 | 0 | 16 | 11 | 0 |
| | 6,2 | 17 | 0 | 12 | 13 | 18 | 10 |
| | 1,3 | 0 | 0 | 0 | 19 | 0 | 11 |
| | 2,3 | 0 | 0 | 10 | 0 | 15 | 17 |
| | 3,3 | 15 | 0 | 0 | 18 | 13 | 0 |
| | 4,3 | 0 | 0 | 0 | 0 | 14 | 17 |
| | 5,3 | 13 | 11 | 20 | 20 | 16 | 19 |
| | 6,3 | 0 | 0 | 0 | 0 | 19 | 20 |
| | 7,3 | 18 | 0 | 11 | 11 | 18 | 20 |

| Jobs | Op. | Machines | | | | | |
|---|---|---|---|---|---|---|---|
| 3.63 | k,p | M1 | M2 | M3 | M4 | M5 | M6 |
| | 1,1 | 0 | 0 | 18 | 0 | 14 | 0 |
| | 2,1 | 0 | 0 | 0 | 20 | 16 | 17 |
| | 3,1 | 13 | 14 | 10 | 0 | 17 | 0 |
| | 4,1 | 11 | 0 | 16 | 17 | 14 | 14 |
| | 5,1 | 12 | 13 | 19 | 10 | 16 | 0 |
| | 6,1 | 0 | 19 | 0 | 14 | 0 | 11 |
| | 7,1 | 17 | 19 | 14 | 15 | 11 | 12 |
| 4 | 8,1 | 0 | 17 | 13 | 15 | 0 | 0 |
| | 1,2 | 0 | 0 | 15 | 0 | 0 | 20 |
| | 2,2 | 0 | 15 | 14 | 13 | 20 | 20 |
| | 3,2 | 0 | 15 | 11 | 0 | 0 | 16 |
| | 4,2 | 11 | 0 | 17 | 0 | 12 | 13 |
| | 5,2 | 18 | 10 | 0 | 0 | 0 | 19 |
| | 6,2 | 0 | 11 | 0 | 0 | 10 | 0 |
| | 7,2 | 15 | 17 | 15 | 0 | 0 | 18 |
| | 8,2 | 13 | 0 | 0 | 0 | 0 | 0 |
| | 1,1 | 14 | 17 | 13 | 11 | 20 | 20 |
| | 2,1 | 16 | 19 | 0 | 0 | 0 | 0 |
| | 3,1 | 19 | 20 | 18 | 0 | 11 | 11 |
| | 4,1 | 18 | 20 | 0 | 0 | 18 | 0 |
| | 5,1 | 14 | 0 | 0 | 0 | 0 | 20 |
| | 6,1 | 16 | 17 | 13 | 14 | 10 | 0 |
| | 1,2 | 17 | 0 | 11 | 0 | 16 | 17 |
| | 2,2 | 14 | 14 | 12 | 13 | 19 | 10 |
| | 3,2 | 16 | 0 | 0 | 19 | 0 | 14 |
| | 4,2 | 0 | 11 | 17 | 19 | 14 | 15 |
| 5 | 1,3 | 11 | 12 | 0 | 17 | 13 | 15 |
| | 2,3 | 0 | 0 | 0 | 0 | 15 | 0 |
| | 3,3 | 0 | 20 | 0 | 15 | 14 | 13 |
| | 4,3 | 20 | 20 | 0 | 15 | 11 | 0 |
| | 5,3 | 0 | 16 | 11 | 0 | 17 | 0 |
| | 6,3 | 12 | 13 | 18 | 10 | 0 | 0 |
| | 1,4 | 0 | 19 | 0 | 11 | 0 | 0 |
| | 2,4 | 10 | 0 | 15 | 17 | 15 | 0 |
| | 3,4 | 0 | 18 | 13 | 0 | 0 | 0 |
| | 4,4 | 0 | 0 | 14 | 17 | 13 | 11 |
| | 5,4 | 20 | 20 | 16 | 19 | 0 | 0 |
| | 1,1 | 0 | 0 | 19 | 20 | 18 | 0 |
| | 2,1 | 11 | 11 | 18 | 20 | 0 | 0 |
| | 3,1 | 18 | 0 | 14 | 0 | 0 | 0 |
| 6 | 4,1 | 0 | 20 | 16 | 17 | 13 | 14 |
| | 5,1 | 10 | 0 | 17 | 0 | 11 | 0 |
| | 6,1 | 16 | 17 | 14 | 14 | 12 | 13 |
| | 7,1 | 19 | 10 | 16 | 0 | 0 | 19 |
| | 1,2 | 0 | 14 | 0 | 11 | 17 | 19 |

*Table A.45 Costs PP3 FJSP-2F instance 6 x 6 with 111 operations (continued).*

| Jobs | Op. | Machines | | | | | |
|---|---|---|---|---|---|---|---|
| 3.63 | k,p | M1 | M2 | M3 | M4 | M5 | M6 |
| | 2,2 | 14 | 15 | 11 | 12 | 0 | 17 |
| | 3,2 | 13 | 15 | 0 | 0 | 0 | 0 |
| | 4,2 | 15 | 0 | 0 | 20 | 0 | 15 |
| 6 | 5,2 | 14 | 13 | 20 | 20 | 0 | 15 |
| | 6,2 | 11 | 0 | 0 | 16 | 11 | 0 |
| | 7,2 | 17 | 0 | 12 | 13 | 18 | 10 |
| | 8,2 | 0 | 0 | 0 | 19 | 0 | 11 |

# VITA AUCTORIS

NAME: Alejandro Vital Soto

PLACE OF BIRTH: Puebla, Mexico

YEAR OF BIRTH: 1985

EDUCATION: B.Sc. in Industrial Engineering, Meritorious Autonomous University of Puebla, Mexico, 2008.

M.Sc. in Industrial Engineering, University of the Americas Puebla, Mexico, 2011.

PUBLICATIONS: Journal Papers

1. Vital Soto, Alejandro, et al. "Mathematical modeling and hybridized evolutionary LP local search method for lot-sizing with supplier selection, inventory shortage, and quantity discounts." Computers & Industrial Engineering 109 (2017): 96-112.