INSTITUTO POLITÉCNICO DE BRAGANÇA
Escola Superior de Tecnologia e Gestão

MODELLING AND OPTIMIZATION OF FLEXIBLE MANUFACTURING SYSTEMS

Víctor Hugo Granados Fernández

Final Project Dissertation presented to

Escola Superior de Tecnologia e Gestão

Instituto Politécnico de Bragança

For obtaining the degree of Master in

Industrial Engineering

Area of **Electrical Engineering**

July 2012

MODELLING AND OPTIMIZATION OF FLEXIBLE MANUFACTURING SYSTEMS

Víctor Hugo Granados Fernández

Final Project Dissertation presented to

Escola Superior de Tecnologia e Gestão

Instituto Politécnico de Bragança

For obtaining the degree of Master in

Industrial Engineering

Area of **Electrical Engineering**

Supervisors:

Prof. Dra Ana Isabel Pereira

Prof. Dr. Paulo Jorge Pinto Leitão

"This dissertation does not include the reviews made by the jury"

July 2012

Dedication

A mis padres, hermana y a Dovilė, que siempre creyeron en mí.

Acknowledgments

First of all, I want to thank the Erasmus Mundus E2NHANCE program for giving me the opportunity to enrich my professional knowledge and cultural experience.

I also offer my sincerest gratitude to my adviser, Dra Ana Pereira, who has supported me throughout the develpoment of my thesis with her patience and guidance.

Dr. Paulo Leitão, who has been an excellent and dynamic Master's director, always caring about this foreign student.

TEMPO-PSI team in France, particularly to Dr. Damien Trentesaux who received me with open arms. Dr. Abdelghani Bekrar, who pointed me in the right direction during this investigation and to Eng. Gabriel Zambrano Rey, who helped this alien to adapt on a different planet. More than a colleague, a friend.

To my family who has believed in me all my life, my accomplishments are their glories.

To Eng. Mario Saenz, whose constant support have enriched my confidence towards this investigation.

To Vitor Laranjeira, a brother-like friend who made me feel at home during my years in Portugal.

And last but not least, to all my friends, whom always wish the best for me. Thank you all.

Abstract

The Lagrangian relaxation method for solving a Mixed-Integer Linear Problem was implemented in order to improve the current TEMPO-PSI team flexible manufacturing system solutions, available at the AIP-primeca pole in University of Valenciennes, France.

The MILP optimization model, from TEMPO-PSI team, can not guarantee good results when the amount of jobs in the manufacturing system increases due to the growth in the number of variables of the model.

A study on the optimization model was conducted with the objective to find the candidate constraints for Lagrangian relaxation. A comparative study is made between the initial model solutions and Lagrange-relaxed model solutions.

The results of the present research indicate that the use of the Lagrangian relaxation method on the scheduling problems could have advantages in obtaining better solutions.

Keywords: Flexible manufacturing systems, Lagrangian relaxation, Mixed-Integer Linear Problems, Scheduling.

Resumo

Um método relaxação lagrangeana para resolver um problema de Programação Misto-Inteira Linear foi implementado com o objetivo de melhorar as soluções actuais dos sistemas de manufatura flexíveis da equipa TEMPO-PSI, disponível no laboratório AIP-primeca da Universidade de Valenciennes, França.

O modelo de optimização MILP, da equipa TEMPO-PSI, não consegue garantir bons resultados quando a quantidade de operações no sistema de manufactura aumenta, devido ao crescimento do número de variáveis do modelo.

O estudo do modelo de optimização foi levado a cabo com o objectivo de encontrar as restrições candidatas para a relaxação Lagrangeana. O estudo comparativo é feito entre as soluções iniciais do modelo e as soluções do modelo relaxado com o método de Lagrange.

Os resultados da presente investigação indicam que o uso de método da relaxação Lagrangeana nos problemas de escalonamento de tarefas poderá ter vantagens na obtenção de melhores soluções.

Palavras-chave: Sistemas de Manufatura flexíves, Relaxação Lagrangeana, Problemas de Programação Inteira-Mista Linear, Escalonamento de tarefas.

Nomenclature

AIP Atelier Inter-Etablissments de Productique (Workshop Inter-Integrated Manufacturing Facilities)

FJSP Flexible Job-Shop Problem

FMS Flexible Manufacturing System

IPB Instituto Politécnico de Bragança

JSP Job-Shop Problem

MILP Mixed-Integer Linear Problem

TEMPO-PSI Thermique Ecoulement Mécanique Matériaux Mise en Forme Production - Production Service Information, research team, University of Valencienes.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The requirements on production of today's industry have increased not only in volume or precision, but also in variety and efficiency. According to Bousbia [3]. "we have noticed an inherently related evolution of the needs expressed by the industrialists."

Every time more and more complex automated manufacturing systems are producing what we consume in our everyday life.

A manufacturing system could be described as an orchesta, on which every instrument plays a different role in the production of the final product, the symphony. The automatic control of a manufacturing system can be analog to the orchesta's director, whose role is to coordinate every movement of the musicians during the function. This coordination gets more and more complicated when the amount of instruments on the orchesta increases, among other factors, such as: delays, breakdowns, order modifications and more.

Mathematical models are typically used to study the behavior of manufacturing systems. The difficulty to solve those models radicates into their inherent high combinational nature, where the amount of variables of the model increases significantly with the growth of the products to be produced, this introduces a bigger computational weight during the process of finding solutions to the model.

When an orchesta gets too complicated (or big) to direct, as well as the mathematical modelling of a manufacturing system gets too complicated to solve, it is necessary to make some adjustments on the symphony so the final tone is similar to the original, yet easier to direct. The same simplification is what mathematical relaxation pursues. Aproximate but easier and feasible solutions to a very complicated problem.

The manufacturing systems nowadays are seeking for flexibility on their productions because it provides a higher and faster adaptability and robustness against perturbations. A notion on flexible manufacturing systems is presented on Chapter 2.

1.1 Objectives

The development of this research has the following objectives:

- To obtain optimal and feasible solutions in shorter CPU processing time applying mathematical relaxation to the MILP model designed by TEMPO-PSI team.
- To compare the relaxed with the un-relaxed model solutions to evaluate the advantages and disadvantages of the mathematical method.
- To provide TEMPO-PSI team with solutions to their MILP model for different scenarios in the system.

1.2 Organization

In chapter 2, different kinds of schedulings for FMS systems are presented, along with their problems and the solution methods commonly used.

The Chapter 3 characterizes the study case, conformed by a flexible manufacturing system available at the University of Valenciennes, France. Its description and real implementation are both detailed to fully understand its functionality.

The Chapter 4 consists on a mathematical modeling of the study case. A comparative study of the results given by the initial model vs the relaxation of constraints is presented so the advantages and disadvantages of each model are highlighted.

Chapter 5 analyzes the results generated by the application of Lagrangian relaxation in comparison with the previous results found by TEMPO-PSI team.

In Chapter 6, the conclusions and future work describe the importance of the findings upon this research.

Chapter 2

Scheduling problem in the FMS

2.1 Flexible Manufacturing Systems.

Several definitions on the term Manufacturing Flexibility have been presented. According to Gupta and Goyal [8], manufacturing flexibility is "the ability of a manufacturing system to cope with changing circumstances or instability caused by the environment". Swamidass [19] defines it as "the capacity of a manufacturing system to adapt successfully to changing environmental conditions as well as changing product and process requirements".

In general, Flexible Manufacturing Systems (FMS) are those which can adapt over certain kinds of changes in the process enviroments, being able to continue their functions successfully with a certain modified performance.

Althought there are several kinds of manufacturing flexibilites, the two basic types are highlighted on this investigation since they are needed by the rest (Product, Process, Operation, Volume, Expansion and Production Flexibility) [18]. See Figure 2.1.

1. Machine Flexibility: Several operations can be performed by the same machine.

2. Routing Flexibility: A given operation can be performed by several machines.

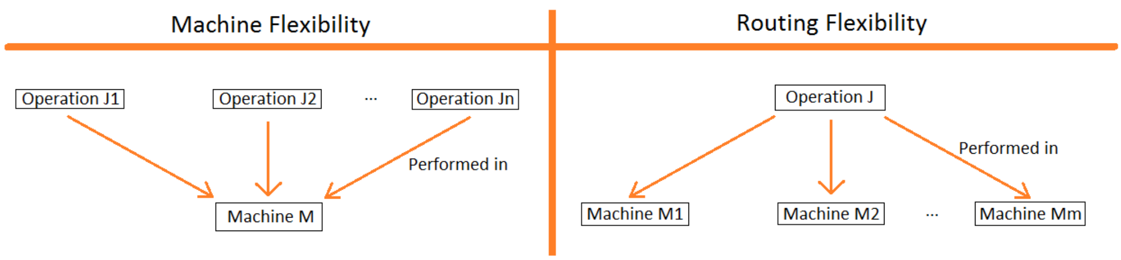


Figure 2.1: Machine Flexibility vs Routing Flexibility

The FMS scheduling problem consists in coordinating the production of every operation of every product performed in the system. "The objectives can be the minimisation of the completion time of jobs, mean flow time, lateness of jobs, processing cost, etc" [14].

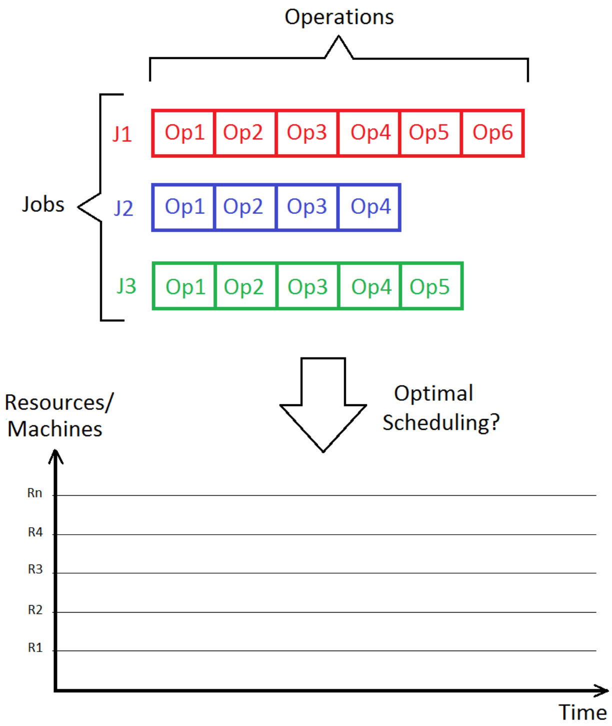


Figure 2.2: Scheduling problem in a FMS.

According to [4]: "FMS scheduling has received a great deal of attention because reducing lead time is always a very important goal for industry".

At a hardware level, the FMS's have the characteristics of possessing programmable machinery for multi purposes, this makes them easy to re-configure to operate in different flexible production cells.

According to [12], a FMS implementation has the following features:

Advantages

- Faster, lower-cost changes from one part to another which will improve capital utilization.
- Lower direct labor cost, due to the reduction in number of workers.
- Reduced inventory, due to the planning and programming precision.
- Consistent and better quality, due to the automated control.
- Lower cost/unit of output, due to the greater productivity using the same number of workers.
- Savings from the indirect labor, from reduced errors, rework, repairs and rejects.

Disadvantages

- Limited ability to adapt to changes in product or product mix (ex. machines are of limited capacity and the tooling necessary for products, even of the same family, is not always feasible in a given FMS).
- Substantial pre-planning activity.
- Expensive, costing millions of dollars.
- Technological problems of exact component positioning and precise timing necessary to process a component.
- Sophisticated manufacturing systems.

The biggest advantage of flexible manufacturing systems over other manufacturing configurations is the adaptability and thus, continuity of production over certain eventualities applied to the system. This adaptation feature will depend directly from each case and how robust each component is, towards different perturbations. No matter how complex and robust a FMS might be, it can not foresees every possible eventuality. However, most FMS are designed to react favorably to the most common perturbations know due to the process characteristics.

The redundancy in equipment capacities, routing combinations, among others, combine with more robust and demanding production requirements, make the Flexible Manufacturing Systems a new tendency in terms of productions structures. However, this new production systems are very complex in terms of modelling, optimization and control due to its highly combinational nature, in fact they have been proved to be NP-hard problems [5].

For example: A FMS with 12 jobs to be processed, each job consisting of 10 operations and 5 workstations where different operations can be performed. This conditions mean a total of 120 operations to be scheduled on the 5 workstations over the time.

Looking at only one workstation, in which 3 out of 5 operations can be performed, (meaning around 72 possible operations to be performed by this workstation) there are $72! = 6.12344584 \times 10^{103}$ possible ways of scheduling the operations over the time. Of course this rough calculation also needs to take in count the rules of operation sequences to discard the non-feasible permutations. The amount of posible schedules depends factorially of the number of operations to be performed.

The inclusion of more jobs to performed by the FMS makes the problem even more difficult to solve because it increases the computational weight factorially, causing a combinatorial explosion (See [9]) if the problem is attempted to be solved by exhaustive search.

Although nowaday's computers have higher computing capacities, it is still difficult to obtain fast solutions for the problem. In fact, parallel computer processing is also being used in scheduling problems in order to reduced its solving time, but it increases the costs of planning and implementation. (For further reading about this topic see [6]).

Commonly, there are three categories used for solving these kind of problems in which all the methods can be divided [15]:

1. Optimization methods: Optimal solutions, require big amount of iterations and computer time. Faster solutions are found when allowing the objective not to be the optimal.

2. Heuristic methods: Based on dispatching rules with good results, however, every system requiers different configurations, which consumes big amount of time. The genetic algorithm is one of the most common techniques on this category.

3. Hybrid methods: Combine the advantages of other methods, usually exploring the state spaces with heuristics. Difficult to implement due to the huge times in planning and modeling each systems.

Nowadays computers have increased the possibilities to find optimal solutions for static models of FMS, however, it is still difficult and unpractical to find solutions for systems with medium-high number of products. The number of variables involved in the optimization process increasses dramatically with the introduction of products on a FMS.

Different techniques for finding optimal solutions (or lower bounds) are used in order to have practical and feasible solutions for a particular difficult problem (More on this topic can be found in [13]). On industries, most manufacturing systems operate in a dynamic enviroment with real-time events which usually turns static-predictive optimal scheduling techniques unstable or unfeasible. However, static scheduling techniques such as optimization of mixed-integer linear problem (MILP) models gives a framework to other approaches due to the optimal nature of its solutions.

Accordin to [14]: Dynamic scheduling techniques for control purposes such as dispatching rules, heuristics, metaheuristics, knowledge-based systems, fuzzy logic, neural networks, Petri nets, hybrid techniques, and multiagent systems have a common characteristic: Its solutions are practical and adaptable but most of the times not optimal.

Static MILP modelling for FMS's give optimal (and thus ideal) solution of the problem in a global approach. This base is used by several dynamic scheduling techniques for comparison and even as an ideal "seed" as a starting point for their solutions.

A MILP model gets more difficult to solve as the real system's size increases. In most of the cases the system can not be solved by analytical methods, exhaustive search, or even linear programing due to its high number of variables. Then several alternatives emerge such as Lagrangian relaxation, penalty methods, linear programing relaxation, among others.

Mathematical relaxation, althought is an uncommon static technique applied to this kind of problems, it is capable to solve them consistently when some degree of versatility is permitted within its constraints. It is the particular case of Lagrange Relaxation that calls the interest of the present investigation.

A flexible assembly cell was used as a case study for modelling, optimization and Lagrangian relaxation of a mixed-integer linear (MILP) problem. The description of said FMS is given in the following Chapter.

Chapter 3

Caracterization of the case study

An application of Lagrangian relaxation was studied in a real case study, a flexible assembly cell at the Valenciennes AIP-Primeca pole of the Univeristy of Valenciennes, France was employed. This Chapter fully describes the cell and its implementation details. Figures and most descriptions from this Chapter are taken from [2].

3.1 AIP-Cell description

3.1.1 Description of the resources and the transportation system

The assembly cell is comformed by seven workstations (W_i), each one of them is placed around a flexible transportation system (See Figures 3.1 and 3.4):

Workstation W_1: This workstation is in charged of loading the base plates onto the shuttles, also performs visual control and unloads the product out of the manufacturing cell.

Workstations W_2, W_3 and W_4: These three workstations use Kuka robots [10] to assemble different components automatically. Each robot (designated as Robot 1, 2 and 3 on Figure 3.1) is capable to assemble three types of components on this FMS. Each component can be picked from two different workstations. This redundancy introduces flexibility into the system, which is particularly useful in cases of robot malfunctions (See Figure 3.2).

Workstation W_5: This workstation is an automated visual control station, taking images of the products and verifying the state of finished products.

Workstation W_6: This workstations is a manual inspection unit. Not used on this implementation.

Workstation W_7: Consists in a new workstation to take several functions to define on future configurations. Not used on this implementation.

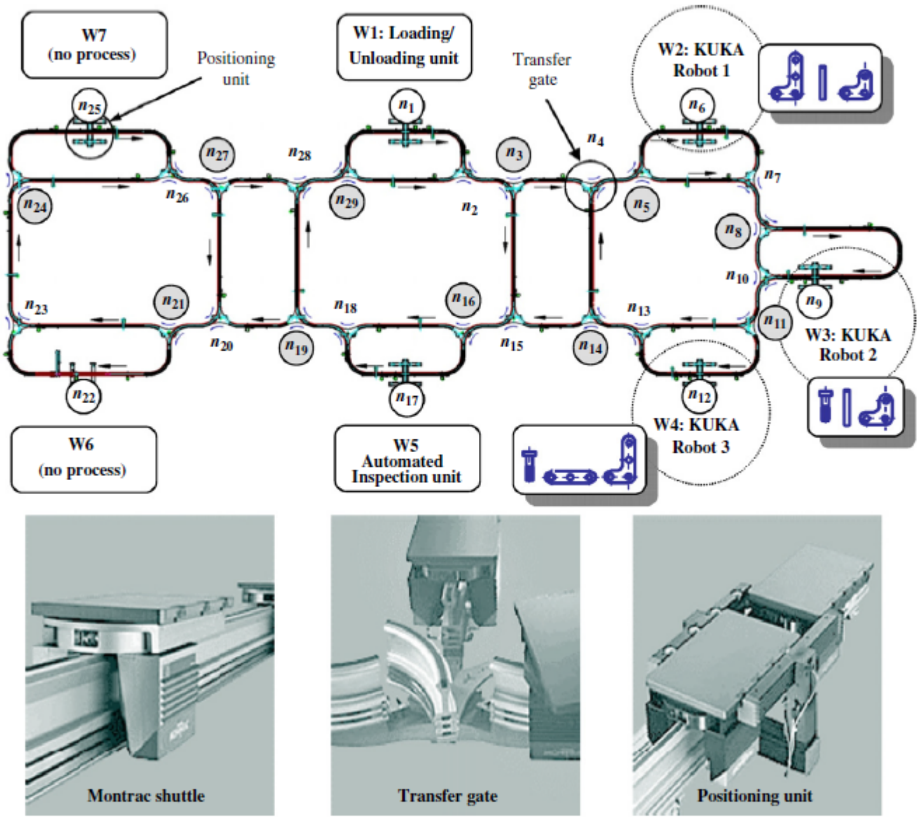


Figure 3.1: Schematic view of the flexible cell

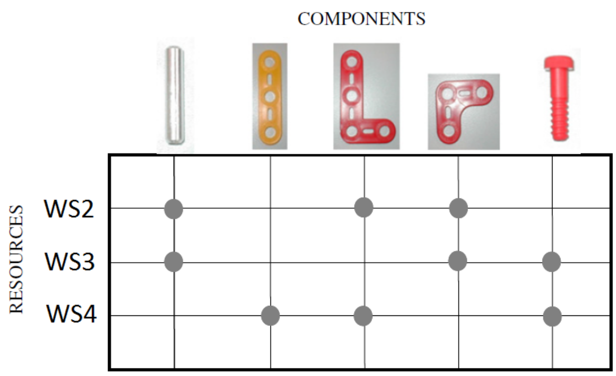


Figure 3.2: Redundancy of workstations/robots and components.

The Table 3.1 shows the manufacturing processing times needed for every workstation to performe each operation of which it is capable. Data taken from [20].

The Table 3.2 shows the transportation times between every adjacent node of the system (taken from [20]), meaning every possible destination node that can be reached directly from a given source node. The nodes are graphically designated over the manufacturing cell on Figure 3.1.

The transportation system is based on modular conveyor components form Montech technology[11]. This system is basically a monorail-like flexible transport system using self-propelled shuttles to transport materials on tracks from one point to another.

	Workstation				
Operation	WS1	WS2	WS3	WS4	WS5
Loading	10	-	-	-	-
Unloading	10	-	-	-	-
Axis_comp	-	20	20	-	-
r_comp	-	20	20	-	-
I_comp	-	-	-	20	-
L_comp	-	20	-	20	-
screw_comp	-	-	20	20	-
Inspection	-	-	-	-	5

Table 3.1: Operations processing times (in seconds)

Destination node

Source node	n1	n2	n3	n4	n5	n6	n7	n8	n9	n10	n11	M1	M2	M3	M4	M5	M6	M7
n1		**4**	-	-	-	-	-	-	-	**5**	-	-	-	-	-	-	-	-
n2	-		**4**	-	-	-	-	-	-	-	-	**5**	-	-	-	-	-	-
n3	-	-		**4**	-	-	-	**5**	-	-	-	-	-	-	-	-	-	-
n4	-	-	-		**4**	-	-	-	-	-	-	-	**5**	-	-	-	-	-
n5	-	-	-	-		**3**	-	-	-	-	-	-	-	**11**	-	-	-	-
n6	-	-	-	-	-		**4**	-	-	-	-	-	-	-	**5**	-	-	-
n7	-	-	-	**5**	-	-		**4**	-	-	-	-	-	-	-	-	-	-
n8	-	-	-	-	-	-	-		**4**	-	-	-	-	-	-	**5**	-	-
n9	-	**5**	-	-	-	-	-	-		**4**	-	-	-	-	-	-	-	-
n10	-	-	-	-	-	-	-	-	-		**4**	-	-	-	-	-	**7**	-
n11	**9**	-	-	-	-	-	-	-	-	-		-	-	-	-	-	-	**10**
M1	-	-	-	**6**	-	-	-	**7**	-	-	-		-	-	-	-	-	-
M2	-	-	-	-	-	**5**	-	-	-	-	-	-		**13**	-	-	-	-
M3	-	-	-	-	-	-	**6**	-	-	-	-	-	-		**7**	-	-	-
M4	-	-	-	**7**	-	-	-	**6**	-	-	-	-	-	-		-	-	-
M5	-	**7**	-	-	-	-	-	-	-	**6**	-	-	-	-	-		-	-
M6	**12**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		**13**
M7	-	**6**	-	-	-	-	-	-	-	**7**	-	-	-	-	-	-	-	

Table 3.2: Transportation times between nodes (in seconds)

Based on the system's configuration, a graph conformed by nodes can be used to model the parts of the manufacturing system. There are three kinds of nodes, called resource, divergent and transfer nodes, they are detailed on the Table 3.3:

3.1.2 Product description

The catalogue of product produced in the manufacturing systems consists of assembling seven types of jobs, each one of them is an alphabet letter: ‘‘b’’, ‘‘E’’, ‘‘L’’, ‘‘T’’, ‘‘A’’, ‘‘I’’ and ‘‘P’’.

Every job is made out of five different components: ‘‘Axis_comp’’, ‘‘I_comp’’, ‘‘L_comp’’, ‘‘r_comp’’ and ‘‘screw_comp’’. In order to build a job, several components need to be assembled over a plate which is mounted on each shuttle. See Figure 3.3.

Node numbers	Type of node	Characteristics
n_1, n_6, n_9, n_{12}, n_{17}, n_{22}, n_{25}	Resource	Nodes that offer services, also called destination nodes
n_3, n_5, n_8, n_{11}, n_{14}, n_{16}, n_{19}, n_{21}, n_{24}, n_{27}, n_{29}	Divergent	Decision nodes, based on transfer gates to direct shuttles on the tracks
n_2, n_4, n_7, n_{10}, n_{13}, n_{15}, n_{18}, n_{20}, n_{23}, n_{26}, n_{28}	Transfer	Consist of transfer gates without decisions

Table 3.3: AIP-Cell node description

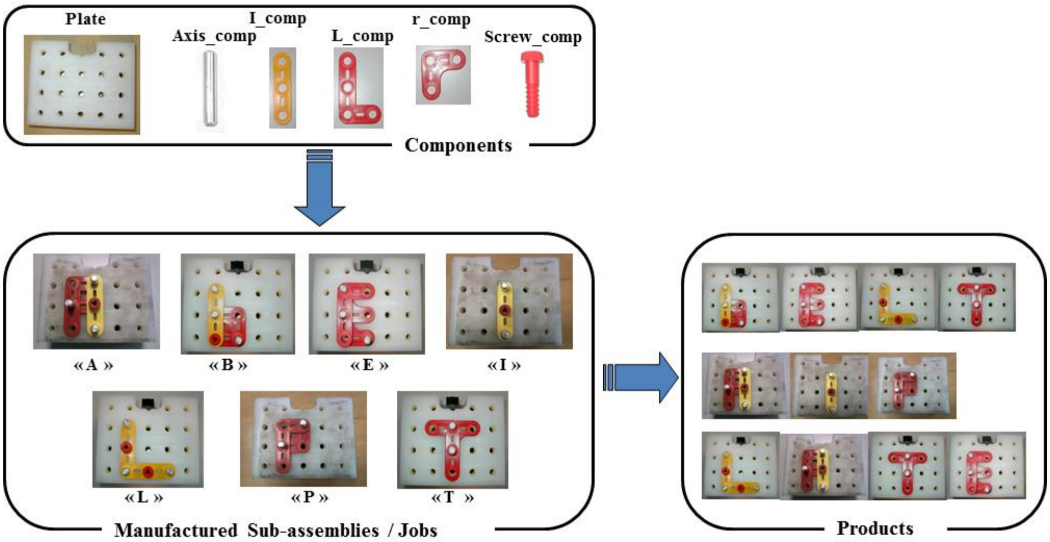


Figure 3.3: Components, Jobs and products.

It is relevant to point out that for each job of the same kind to be sucessfully assembled, the order of mounting and placing the components has to be the same as the ones indicated in the Table 3.4. This production requirement guarantees homogeneity of the final products.

The Figure 3.4 shows a real view of the flexible manufacturing system study case The workstations are also indicated.

The AIP-Cell is currently being used as an experimental study case by the TEMPO-PSI team and for teaching purposes to students of University of Valenciennes. The scheduling problem of this manufacturing cell by static modelling and optimization has proved to be difficult to achieve when the number of jobs in the systems increases until seven or more. On the next Chapter, a Lagrangian relaxation was applied to the MILP model of the cell in order to obtain better results.

Process	“b” product	“E” product	“L” product	“T” product	“A” product	“I” product	“P” product
Phase 1	Loading	Loading	Loading	Loading	Loading	Loading	Loading
Phase 2	Axis_comp	Axis_comp	Axis_comp	Axis_comp	Axis_comp	Axis_comp	Axis_comp
Phase 3	Axis_comp	Axis_comp	Axis_comp	Axis_comp	Axis_comp	Axis_comp	Axis_comp
Phase 4	Axis_comp	Axis_comp	Axis_comp	L_comp	Axis_comp	I_comp	r_comp
Phase 5	r_comp	r_comp	I_comp	r_comp	r_comp	screw_comp	L_comp
Phase 6	r_comp	r_comp	I_comp	Inspection	L_comp	Inspection	Inspection
Phase 7	I_comp	L_comp	screw_comp	Unloading	I_comp	Unloading	Unloading
Phase 8	screw_comp	Inspection	screw_comp		screw_comp		
Phase 9	Inspection	Unloading	Inspection		Inspection		
Phase 10	Unloading		Unloading		Unloading		

Table 3.4: Assembly process plan for each product.

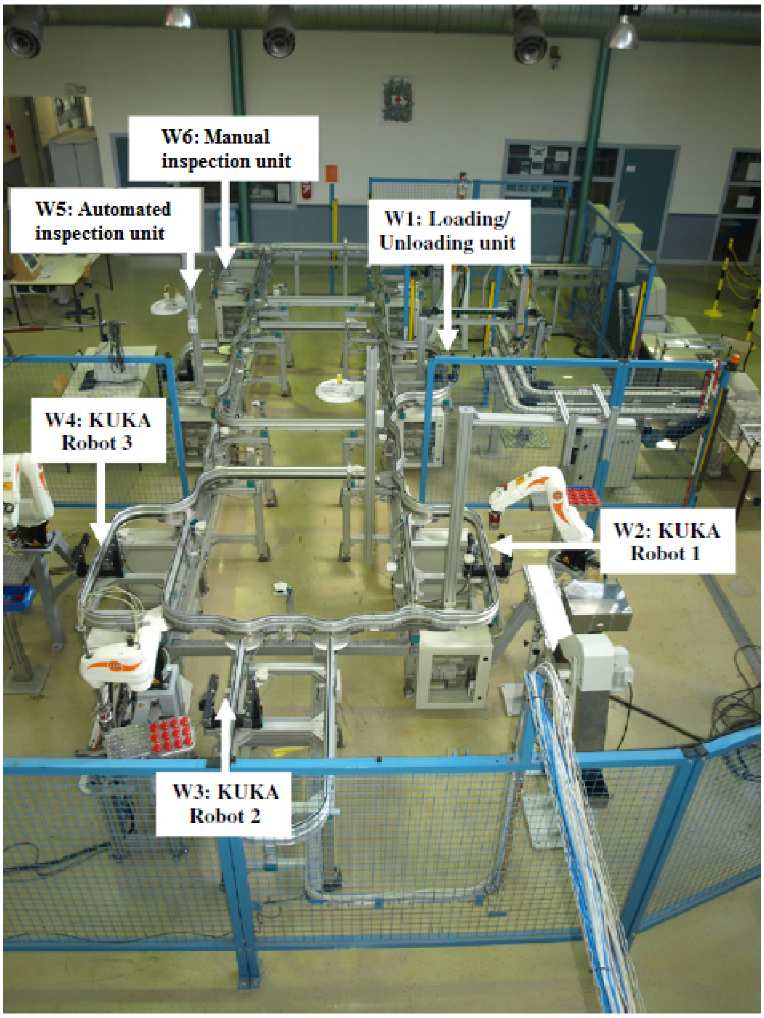


Figure 3.4: Real view of the flexible cell

Chapter 4

Aplication of Lagrangian relaxation to the scheduling problem

4.1 TEMPO-PSI Formulation

The University of Valenciennes, in France, specifically the TEMPO-PSI team has developed a model for the FMS described on the study case of Chapter 2 [2].

For the resolution of the FJSP it is considered a mixed-integer linear program (MILP) model that represents the mathematical formulation of the scheduling problem.

According to its developers [2]: "The results of the MILP can only be considered as a lower bound since all the real constraints cannot be taken into account in this formal model". This consideration takes in count that every model, of any given system, can not represent its properties and functions on every single way, yet it gets close to do it.

4.1.1 Formulation

The objective of this MILP is to minimize the completion time of all the production operations:

$$C_{max} = max(t_{ij}) \quad \forall i \in I_j, \forall j \in P \tag{4.1}$$

J	Set of jobs, $J = 1, 2, \ldots n$
P	set of jobs to be performed: $P = 1, 2, \ldots n$, denoted by the expression $j \in P$
R	Set of machines, $R = 1, 2, \ldots r$
I_j	Set of operation of the job j, $Ij = 1, 2, \ldots, \lvert Ij \rvert$, $j \in J$
O_{ij}	Operation i of the job j
M_{ij}	Set of possible machines for the operation O_{ij}
p_{ij}	Processing time of operation i $(i \in I_j)$
tt_{m1m2}	Transportation time from machine m_1 to m_2
MJ	Maximum simultaneous jobs in the shop floor
ci_r	Input queue capacity of machine r
d_j	Due date of job j, j=1, ..., n
α_j ,β_j	Are the tardiness and lateness penalties associated to the job j, j=1, ..., n

Notations for variables

t_{ij}	completion time of operation O_{ij} $(i \in I_j)$,$t_{ij} \in N$.
μ_{ijr}	a binary variable set to 1 if operation O_{ij} is performed on machine r, 0 otherwise.
b_{ijkl}	a binary variable set to 1 if operation O_{ij} is performed before operation O_{kl}, 0 otherwise.
tr_{ijr1r2}	a binary variable set to 1 if job j is transported to machine r_2 after performing operation O_{ij}, 0 otherwise.
w_{ijr}	waiting time of operation O_{ij} in the queue of machine r.
wv_{ijklr}	a binary variable set to 1 if operation O_{ij} is waiting for operation O_{kl} in the queue of machine r, 0 otherwise,
z_{lj}	set to 1 if job l and job j are in the shop floor in the same time, 0 otherwise.

Detail of the constraints

Disjunctive constraints: A machine can process one operation at time and an operation is performed by only one machine.

$$t_{ij} + p_{kl} * u_{klr} + BM * b_{ijkl} \leq t_{kl} + BM \quad \forall i, k \in I,\ \forall j, l \in P,\ \forall r \in R_{ij} \tag{4.2}$$

where BM is a large number

$$b_{ijkl} + b_{klij} \leq 1 \quad \forall i \in I_j,\ k \in I_l,\ \forall j, l \in P \tag{4.3}$$

$$\sum_{r \in R_{ij}} \mu_{ijr} = 1 \quad \forall i \in I_j,\ \forall j \in P \tag{4.4}$$

Precedence constraints: Ensure the task sequence of a product. The completion time of the next operation considers the completion time of the precedent one, the waiting time and the transportation time if the two operations are not performed in the same machine.

$$t_{(i+1)j} \geq t_{ij} + p_{(i+1)j} + w_{(i+1)jr_2} + \textstyle\sum_{r_1, r_2 \in R} tt_{r_1 r_2} tr_{ijr_1 r_2} \qquad \forall i \in I_j,\ \forall j \in P,\ \forall r_1, r_2 \in R_{ij} \tag{4.5}$$

$$\sum_{\substack{r_1, r_2 \in R \\ r_1 \neq r_2}} tr_{ijr_1 r_2} \leq 1 \quad \forall i \in I_j,\ \forall j \in P \tag{4.6}$$

Allocation and transportation relationship: If successive operations of a product are performed on different machines, this implies that there is a transport operation between those two machines. Transportation delays are set to zero and the transportation system has unlimited capacity.

$$\mu_{ijr_1} + \mu_{(i+1)jr_2} - 1 \leq tr_{ijr_1 r_2} \quad \forall i \in I_j,\ \forall j \in P,\ \forall r_1, r_2 \in R_{ij},\ r_1 \neq r_2 \tag{4.7}$$

$$\mu_{ijr_1} + \mu_{(i+1)jr_2} \geq (1 + \epsilon) tr_{ijr_1 r_2} \quad \forall i \in I_j,\ \forall j \in P,\ \forall r_1, r_2 \in R_{ij},\ r_1 \neq r_2 \tag{4.8}$$

where ϵ is a small number

Queue capacity of the machine input and FIFO rule: each machine has a limited queue capacity. No more operations than this capacity can wait in the queue. The first product arriving in the queue is the first treated.

$$b_{ijkl} + wv_{ijklr} \leq 1 \quad \forall i, k \in I,\ \forall j, l \in P,\ \forall r \in R_{ij} \cap R_{kl} \tag{4.9}$$

$$b_{ijkl} - wv_{ijklr} \geq 0 \quad \forall i \in I_j,\ \forall k \in I_l,\ \forall j, l \in P,\ \forall r \in R_{ij} \cap R_{kl} \tag{4.10}$$

$$wv_{ijklr} + wv_{klijr} \leq 1 \quad \forall i \in I_j,\ \forall k \in I_l,\ \forall j, l \in P,\ \forall r \in R_{ij} \cap R_{kl} \tag{4.11}$$

$$t_{ij} - p_{ij} + BM \cdot b_{ijkl} + BM \cdot wv_{klijr} \leq t_{kl} - p_{kl} - w_{klr} + 2 \cdot BM \qquad \forall i \in I_j,\ \forall k \in I_l,\ \forall j, l \in P,\ j \neq l,\ \forall r \in R_{ij} \cap R_{kl} \tag{4.12}$$

$$w_{klr} \leq \sum_{\substack{i \in I \\ j \in P,\, j \neq l}} p_{ij} wv_{klijr} \quad \forall k \in I_l, \forall l \in P, \forall r \in R_{kl} \tag{4.13}$$

$$\begin{gathered} \mu_{ijr} + \mu_{klr} \geq 2(wv_{ijklr} + wv_{klijr}) \\ \forall i \in I_j, \forall k \in I_k, \forall j, l \in P, \forall r \in R_{ij} \cap R_{kl} \end{gathered} \tag{4.14}$$

$$t_{ij} + BM \cdot b_{ijkl} \leq t_{kl} + BM \quad \forall i \in I_j, \forall k \in I_k, \forall j, l \in P \tag{4.15}$$

$$\begin{gathered} t_{ij} - p_{ij}\mu_{ijr} - w_{ijr} + BM \cdot b_{ijkl} \leq t_{kl} - p_{kl}\mu_{klr} - w_{klr} + BM \\ \forall i \in I_j, \forall k \in I_k, \forall j, l \in P, \forall r \in R \end{gathered} \tag{4.16}$$

$$\sum_{\substack{l \in P,\, k \in I_l \\ l \neq j}} wv_{ijklr} \leq ci_r - 1 \quad \forall i \in I_j, \forall j \in P, \forall r \in R_{ij} \cap R_{kl} \tag{4.17}$$

Limitation of the number of product in the system: The number of simultaneous job in the shop floor can be limited by MJ.

$$\sum_{\substack{l \in P \\ l \neq j}} z_{lj} \leq MJ - 1 \quad \forall j \in P \tag{4.18}$$

$$z_{jl} \geq b_{0luj} + b_{0j0l} - 1 \quad \forall j, l \in P \tag{4.19}$$

$$z_{jl} \leq 1 - b_{0l0j} + b_{ujul} \quad \forall j, l \in P \tag{4.20}$$

$$z_{jl} \geq b_{0l0j} + b_{ujul} - 1 \quad \forall j, l \in P \tag{4.21}$$

Constraints for the type of each variable

$$t_{ij} \geq p_{ij} \quad \forall i \in I_j, \forall j \in P \tag{4.22}$$

$$b_{ijkl} \in \{0, 1\} \quad \forall i \in I_j, \forall j \in P, \forall k \in I_l, \forall l, \in P \tag{4.23}$$

$$tr_{ijr_1r_2} \in \{0, 1\} \quad \forall i \in I_j, \forall j \in P, \forall r_1, r_2 \in R_{ij} \tag{4.24}$$

$$\mu_{ijr} \in \{0,1\} \quad \forall i \in I_j, \forall j, \in J, \forall r \in R_{ij} \tag{4.25}$$

4.1.2 Model Solution

The model developed by TEMPO-PSI team is solved for different scenarios using IBM ILOG Optimizer CPLEX 12.4 (User's manual [7]). This solution are called the "Default" solution.

There are two different experiments conducted on this research. Their difference rely on the transportation times. The first experiment (referred to as Experiment 1) has longer transportation times and it was used to generate the trial for finding values on the Lagrangian parameter λ.

The second experiment (Experiment 2) has lower transportation times, and it was used to apply the Lagrangian relaxation method with the previously chosen λ parameters for several production scenarios.

The TEMPO-PSI team solutions to both experiments are resumed in the Tables 4.2 and 4.1.

	Cmax (s)
CPU Time (s)	Default scenario
300	661
600	602
1800	572
5400	571 (optimal)

Table 4.1: *Cmax* for default case for 7 Jobs BELTAIP, Experiment 1.

Scenario	Cmax (s)	Optimal/ PT (s)	Comment
3 jobs AIP	219	Yes	—
4 jobs BELT	276	Yes	—
6 jobs 2AIP	332	No / 1600	Gap of 9%
8 jobs 2BELT	446	No/1600	Gap of 22%

Table 4.2: *Cmax* for different scenarios, default optimization, Experiment 2.

Each result produces a schedule where the operations of each job are processed in different machines through the time.

The operations of each jobs in the schedule diagrams are represented by the mathematical expresion from equation 4.26.

$$\left(\frac{1}{2}\right) \sin\left(\pi \frac{(t - t_{initial})}{t_{operation}}\right) \tag{4.26}$$

Where:

t The global time when a operation is being performed

$t_{initial}$ Initial time of a given operation,

$t_{operation}$ The processing time of a given operation.

Since equation 4.26, is a $\sin(x)$ function, the resulting graphical representation of any operation will be half-period waves, starting from the initial time to the completion time of each operation. The magnitud of the function is diminish by one half in order to create an equal distance from machine to machine. The previous explanation can be observed on Figure 4.1.

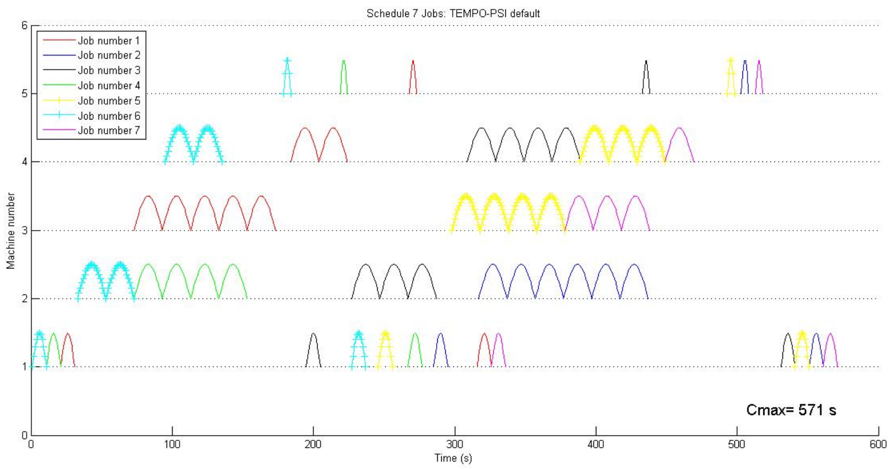


Figure 4.1: Schedule 7 jobs BELTAIP Default case, Experiment 1.

On the Figure 4.1, the optimal schedule for 7 jobs BELTAIP Default-case has a makspan (Cmax) of 571 s (Indicated at the bottom of the image). Each Job is represented by a different color and every operation of each job respects its production sequence (See Table 3.4).

On the next section, the characterization and results of applying the lagrangian relaxation method are presented in order to compare with the current TEMPO-PSI teeam results.

4.2 Lagrangian Relaxation

Lagrangian relaxation is a promising optimization technique for obtaining lower bounds since it allows the MILP model to have a certain degree of freedom on contraints that can be violated parcially. Relaxing constraints in the mathematical formulation could lead to solutions for higher number of jobs (currently no solution for more than 8 jobs), so the TEMPO-PSI team would be enriched with results under these conditions.

An inspection on which constraint(s) could be relaxed (for consistent solution purposes) was conducted with the guide of Dr. Bekrar from TEMPO-PSI team and Dra. Pereira (IPB master thesis guide). It was found out that the model posses two constraints that can be relaxed without loosing feasibility on the final solution. Those constraints are described by equation 4.17 and 4.18. These constraints represent the amount of queue capacity of each workstation and the amount of operations the manufacturing system can performed at any given time, respectively.

On the case of equation 4.17 (called Constraint Ci), the amount of shuttles that can be waiting in the line of any workstation cannot be higher than two. This is due to physical space at the entrance of the workstations. Constraint Ci is flexible for implementation only if some modification is applied to the manufacturing cell.

The equation 4.18 (referred as Constraint MJ) indicates that the system cannot have more than 4 worksations performing tasks during any given time. This constraint is promisingly flexible, since there is no physical implications on the partial implementation of its violation.

4.2.1 Formulation

Consider a linear optimization problem described by the following:

$$\begin{aligned} min \quad & c^T x \\ s.t \quad & Ax \leq b \end{aligned} \tag{4.27}$$

where $x \in \mathbb{R}^n$ and $A \in \mathbb{R}^{m,n}$.

If the constraint $Ax \leq b$ is splitted into two constraints such that $A_1 \in \mathbb{R}^{m_1,n}$, $A_2 \in \mathbb{R}^{m_2,n}$ and $m = m_1 + m_2$, the optimization problem (4.27) can be reformulated as:

$$\begin{aligned} min \quad & c^T x \\ s.t \quad & A_1 x \leq b_1 \\ & A_2 x \leq b_2 \end{aligned} \tag{4.28}$$

The second constraint can be introduced in the objective function.

$$\begin{aligned} min \quad & c^T x + \lambda^T (b_2 - A_2 x) \\ s.t. \quad & A_1 x \leq b_1 \end{aligned} \tag{4.29}$$

Where $L(x, \lambda) \equiv cx + \lambda(b_2 - A_2 x)$ is known as the Lagrange Function, named after the mathematician Joseph-Louis Lagrange (1763-1813). If the weights$\lambda = (\lambda_1, ..., \lambda_{m_2})$ are nonnegative values, then the Lagrange function gets penalized if the included constraint ($A_2 x \leq b_2$) is violated. The objective function also gets rewarded if the constraint is satisfied strictly.

The election of how to split the constraints will depend directly from each problem and the physical meaning of the constraints. Relaxing contraints that are not physically permitted on the problem would lead to have optimal but non-realistic solutions.

The Lagrange relaxation method consists in allowing some constraints not to be fully accomplished, instead it guides the solution to violate those contraints as few times as possible.

Since the MILP model is not derivable, most analitical methods for finding λ do not apply.

The λ vector consists in new set of variables to be determined by during the optimization process. For efects of this research, the λvector was chosen to be constant, several optimization runs were conducted to choose this parameter.

The model presented in the section 4.1.1 has been modified so it can be represented as a relaxed problem with a constraint included inside of the objective function. The notation of the variables stays the same.

For each relaxation, a new model can be described as a new enlarged objective function, plus the rest of the unmodified constraints (obviosly not taking in count the respective relaxed constraint).

Lagrangian relaxation of: Limitation of the maximun number of jobs at the same time in the system.

$$C_{max} = max(t_{ij}) + \lambda_v \left(MJ - 1 - \sum_{\substack{l \in P \\ l \neq j}} z_{lj} \right) \tag{4.30}$$
$$\forall i \in I_j,\ \forall j \in P,\ \forall v \in V_1,\ \forall \lambda \geq 0$$

Where $V_1 = (1, 2, ..., n^2)$, which means that $\lambda = (\lambda_1, ..., \lambda_{n^2})$ and it represents the combinations between jobs in the system. The variable z_{lj} has $n \times n$ dimension.

Lagrangian relaxation of: Queue capacity on each resource (workstation).

$$C_{max} = max(t_{ij}) + \lambda_v \left(ci_r - 1 - \sum_{\substack{l \in P,\ k \in I_l \\ l \neq j}} wv_{ijklr} \right) \tag{4.31}$$
$$\forall i \in I_j,\ \forall j \in P,\ \forall r \in R_{ij} \cap R_{kl},\ \forall v \in V_2,\ \forall \lambda \geq 0$$

On this case, $V_2 = (1, 2, ..., r \times n^2)$, and it represents the combination between jobs and resources in the system. Notice that wv_{ijklr} has $r \times n \times n$ dimension.

The constraints which are not relaxed will stay the intact from the MILP model.

Disjunctive constraints:

$$t_{ij} + p_{kl} * u_{klr} + BM * b_{ijkl} \leq t_{kl} + BM \quad \forall i,k \in I, \forall j,l \in P, \forall r \in R_{ij} \tag{4.32}$$

where BM is a large number

$$b_{ijkl} + b_{klij} \leq 1 \quad \forall i \in I_j,\ k \in I_l,\ \forall j,l \in P \tag{4.33}$$

$$\sum_{r \in R_{ij}} \mu_{ijr} = 1 \quad \forall i \in I_j,\ \forall j \in P \tag{4.34}$$

Precedence constraints:

$$\begin{gathered} t_{(i+1)j} \geq t_{ij} + p_{(i+1)j} + w_{(i+1)jr_2} + \textstyle\sum_{r_1,r_2 \in R} tt_{r_1r_2} tr_{ijr_1r_2} \\ \forall i \in I_j,\ \forall j \in P,\ \forall r_1,r_2 \in R_{ij} \end{gathered} \tag{4.35}$$

$$\sum_{\substack{r_1, r_2 \in R \\ r_1 \neq r_2}} tr_{ijr_1r_2} \leq 1 \quad \forall i \in I_j,\ \forall j \in P \tag{4.36}$$

Allocation and transportation relationship:

$$\mu_{ijr_1} + \mu_{(i+1)jr_2} - 1 \leq tr_{ijr_1r_2} \quad \forall i \in I_j,\ \forall j \in P,\ \forall r_1,r_2 \in R_{ij},\ r_1 \neq r_2 \tag{4.37}$$

$$\mu_{ijr_1} + \mu_{(i+1)jr_2} \geq (1+\epsilon) tr_{ijr_1r_2} \quad \forall i \in I_j,\ \forall j \in P,\ \forall r_1,r_2 \in R_{ij},\ r_1 \neq r_2 \tag{4.38}$$

where ϵ is a small number

Queue capacity of the machine input and FIFO rule:

$$b_{ijkl} + wv_{ijklr} \leq 1 \quad \forall i,k \in I,\ \forall j,l \in P,\ \forall r \in R_{ij} \cap R_{kl} \tag{4.39}$$

$$b_{ijkl} - wv_{ijklr} \geq 0 \quad \forall i \in I_j,\ \forall k \in I_l,\ \forall j,l \in P,\ \forall r \in R_{ij} \cap R_{kl} \tag{4.40}$$

$$wv_{ijklr} + wv_{klijr} \leq 1 \quad \forall i \in I_j,\ \forall k \in I_l,\ \forall j,l \in P,\ \forall r \in R_{ij} \cap R_{kl} \tag{4.41}$$

$$\begin{gathered} t_{ij} - p_{ij} + BM \cdot b_{ijkl} + BM \cdot wv_{klijr} \leq t_{kl} - p_{kl} - w_{klr} + 2 \cdot BM \\ \forall i \in I_j,\ \forall k \in I_l,\ \forall j,l \in P,\ j \neq l,\ \forall r \in R_{ij} \cap R_{kl} \end{gathered} \tag{4.42}$$

$$w_{klr} \leq \sum_{\substack{i \in I \\ j \in P,\, j \neq l}} p_{ij} wv_{klijr} \quad \forall k \in I_l, \forall l \in P, \forall r \in R_{kl} \tag{4.43}$$

$$\begin{gathered} \mu_{ijr} + \mu_{klr} \geq 2(wv_{ijklr} + wv_{klijr}) \\ \forall i \in I_j, \forall k \in I_k, \forall j, l \in P, \forall r \in R_{ij} \cap R_{kl} \end{gathered} \tag{4.44}$$

$$t_{ij} + BM \cdot b_{ijkl} \leq t_{kl} + BM \quad \forall i \in I_j, \forall k \in I_k, \forall j, l \in P \tag{4.45}$$

$$\begin{gathered} t_{ij} - p_{ij}\mu_{ijr} - w_{ijr} + BM \cdot b_{ijkl} \leq t_{kl} - p_{kl}\mu_{klr} - w_{klr} + BM \\ \forall i \in I_j, \forall k \in I_k, \forall j, l \in P, \forall r \in R \end{gathered} \tag{4.46}$$

$$\sum_{\substack{l \in P,\, k \in I_l \\ l \neq j}} wv_{ijklr} \leq ci_r - 1 \quad \forall i \in I_j, \forall j \in P, \forall r \in R_{ij} \cap R_{kl} \tag{4.47}$$

Limitation of the number of product in the system:

$$\sum_{\substack{l \in P \\ l \neq j}} z_{lj} \leq MJ - 1 \quad \forall j \in P \tag{4.48}$$

$$z_{jl} \geq b_{0luj} + b_{0j0l} - 1 \quad \forall j, l \in P \tag{4.49}$$

$$z_{jl} \leq 1 - b_{0l0j} + b_{ujul} \quad \forall j, l \in P \tag{4.50}$$

$$z_{jl} \geq b_{0l0j} + b_{ujul} - 1 \quad \forall j, l \in P \tag{4.51}$$

Constraints for the type of each variable:

$$t_{ij} \geq p_{ij} \quad \forall i \in I_j, \forall j \in P \tag{4.52}$$

$$b_{ijkl} \in \{0, 1\} \quad \forall i \in I_j, \forall j \in P, \forall k \in I_l, \forall l, \in P \tag{4.53}$$

$$tr_{ijr_1r_2} \in \{0, 1\} \quad \forall i \in I_j, \forall j \in P, \forall r_1, r_2 \in R_{ij} \tag{4.54}$$

$$\mu_{ijr} \in \{0, 1\} \quad \forall i \in I_j, \forall j, \in J, \forall r \in R_{ij} \tag{4.55}$$

	Cmax (s)						
CPU	Lagrange Relaxation of MJ Constraint - λ values						Default
Time (s)	[0.1, ..., 0.1]	[0.5, ..., 0.5]	[0.75, ..., 0.75]	[1, ..., 1]	[2, ..., 2]	[20, ..., 20]	
300	467	467	467	497	517	527	661
600	467	466	467	475	466	476	602
1800	456	456	456	467	465	477	572
5400	456	456	456	466	457	475	571 (optim)
9671.73	456	456 (optim)	—	—	—	—	
9792.56	—		456 (optim)	—	—	—	
12064.37	456 (optim)			—	—	—	
14221				—	456	—	
17000				466		466	

Table 4.3: *Cmax* for Relaxed and default cases for 7 Jobs BELTAIP, Experiment 1.

4.2.2 Results

The results from Lagrangian relaxation were obtained using the optimization program IBM ILOG Optimizer CPLEX 12.4, running on a PC with an Intel® Core 2 solo processor @ 1.40 GHz and 4 Gb of RAM memory. These solutions would be called the "Relaxed" solutions.

4.2.2.1 Experiment 1 (longer transportation times)

The Table 4.3 resumes the optimization trials conducted with Lagrangian relaxation of the "MJ" constraint, compared to the "Default" solutions for the Experiment 1 (bigger transportation times).

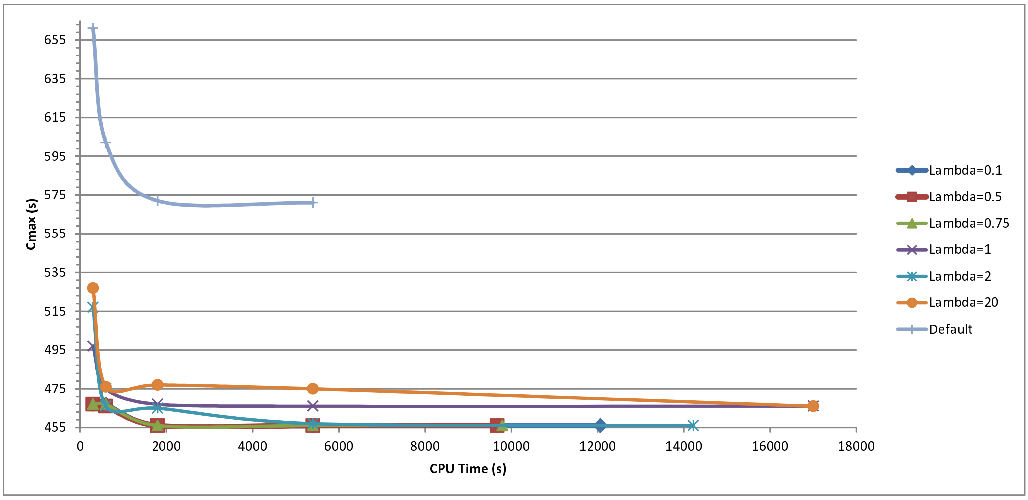
For a better comprehension of the data from Table 4.3, The solution time Cmax was plotted in function of the CPU time for each case of λ and the default case.

It is notorious that, on Experiment 1, all of the Cmax values from all the parameters λ were better than the default case, even when their values are not proved to be optimal by the CPLEX algorithm, they prove to be better than the optimal values of the default case. Having better solutions on shorter CPU processing times present obvious advantages on terms of computer resources, less time for finding solutions and the possibility of applying the method to higher number of jobs included in the flexible manufacturing system.

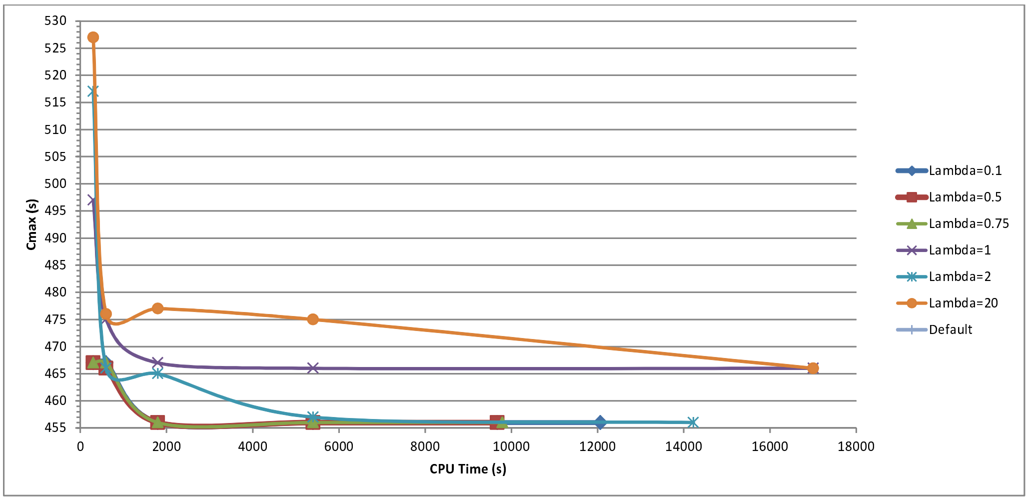
Two λ values are taken from this experiment in order to be used on the next experiment. The best found value is $\lambda = [0.5, ..., 0.5]$ according to the performance described in Figure 4.2 and Table 4.3.

On the case of Lagrangian relaxation of Ci contraints, the Cmax was was plotted in function of the CPU time for each case of λ and the default case. For this Lagrangian relaxation, the parameter λ was chosen to be a vector of 1's and a vector of 2's. The results are compared to the default case on the Figure 4.3.

As seen in Figure 4.3, the Lagangian relaxation of Ci constraint does not imply neither



(a) Lagrangian relaxations vs default case



(b) Comparison of Lagrangian relaxations

Figure 4.2: *Cmax* in function of CPU time for Default and Lagrangian relaxations of MJ constraint, 7 jobs BELTAIP, Experiment 1.

	Cmax (s)			
CPU	Lagrange Relaxation of Ci constraints - λ values			Default
Time (s)	$\lambda = [0.5, ..., 0.5]$	$\lambda = [1, ..., 1]$	$\lambda = [2, ..., 2]$	
300	no solution	651	no solution	661
600	701	601	682	602
1800	591	582	612	572
5400	571	572	571	571 (optimal)

Table 4.4: *Cmax* for relaxed Ci constraints, 7 jobs, Experiment 1.

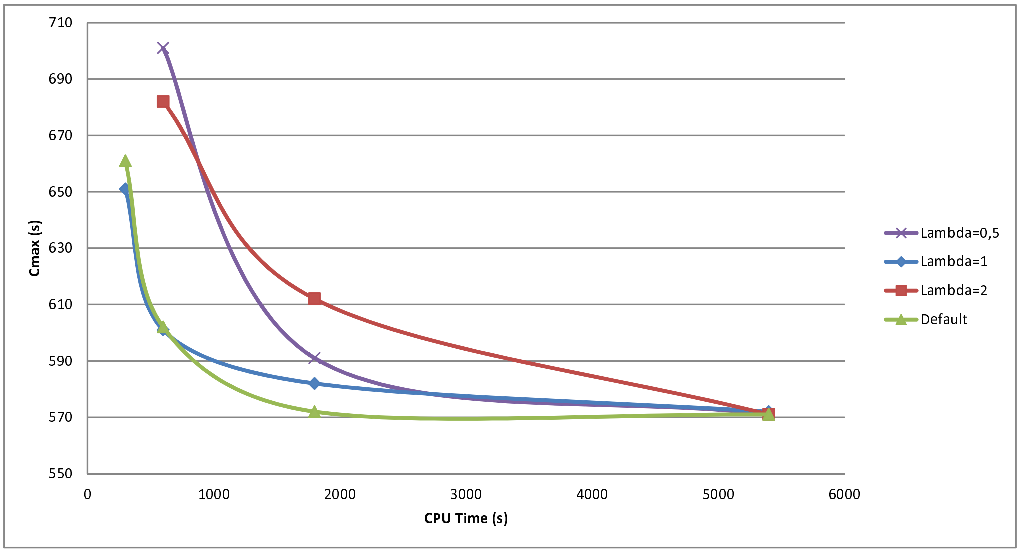


Figure 4.3: Cmax in function of CPU time for Default and Lagrangian relaxations of Ci constraint, 7 jobs BELTAIP, Experiment 1.

a better solution nor smaller CPU processing times for any of the chosen λ parameters. For the experiment 2, this Lagrangian relaxation will not be applied.

The values presented on the Tables 4.3 and 4.4 generate several the schedules depending on the applied parameter λ and the relaxed constraints.

Schedules of Lagrangian relaxation of MJ constraints - 7 jobs BELTAIP - several parameters λ.

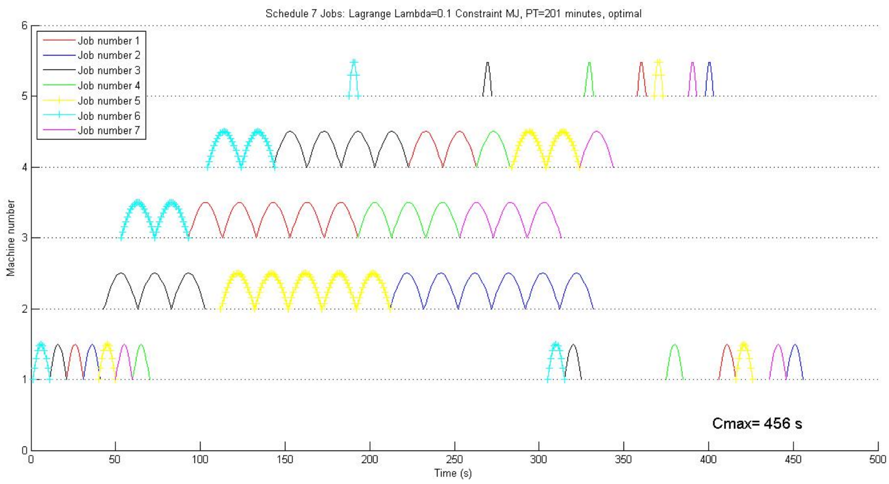


Figure 4.4: Schedule 7 jobs BELTAIP Lagrangian Relaxation MJ constraint, lambda=0.1

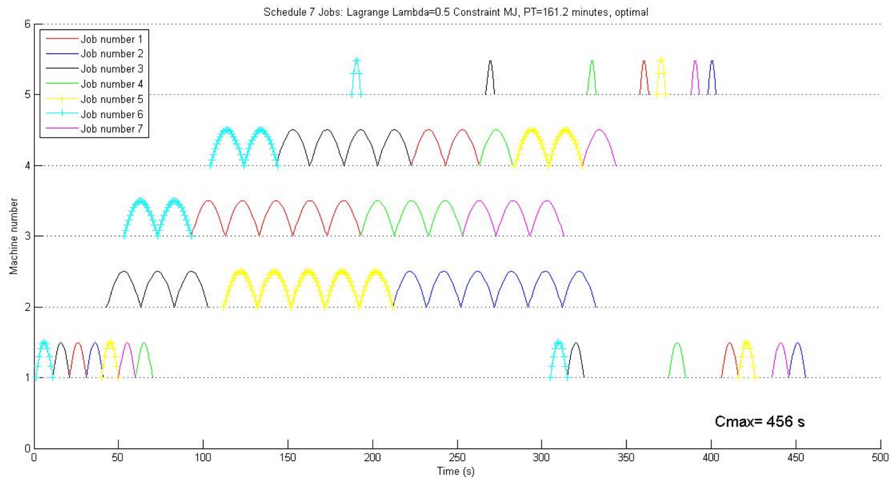


Figure 4.5: Schedule 7 jobs BELTAIP Lagrangian Relaxation MJ constraint, lambda=0.5

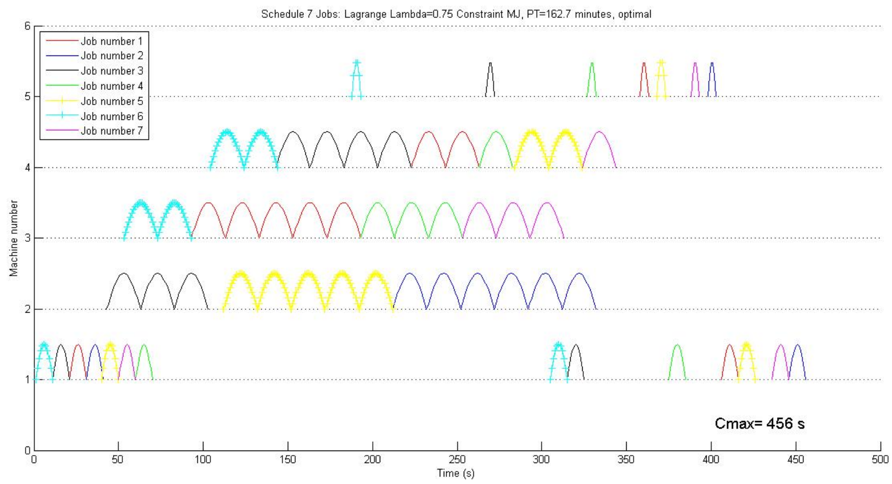


Figure 4.6: Schedule 7 jobs BELTAIP Lagrangian Relaxation MJ constraint, lambda=0.75

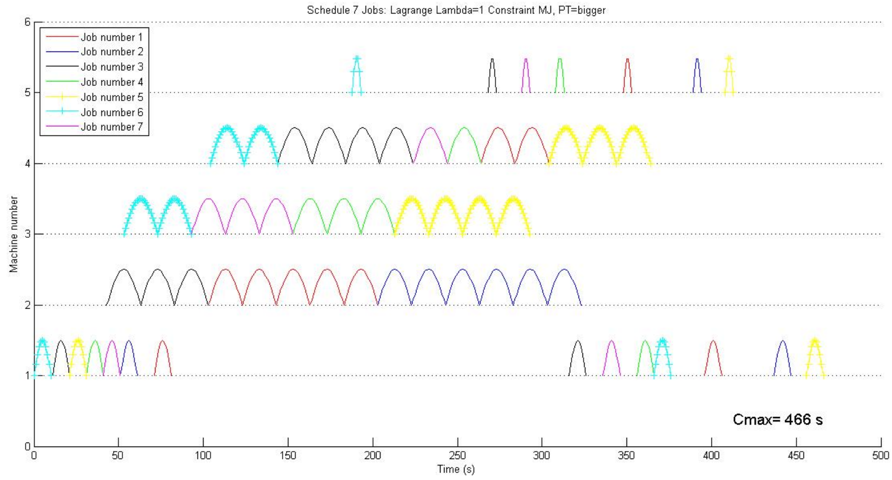


Figure 4.7: Schedule 7 jobs BELTAIP Lagrangian Relaxation MJ constraint, lambda=1

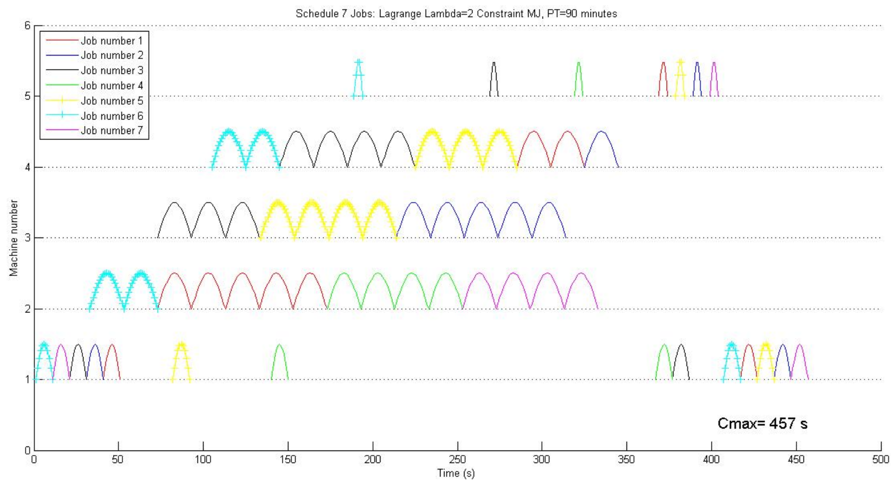


Figure 4.8: Schedule 7 jobs BELTAIP Lagrangian Relaxation MJ constraint, lambda=2

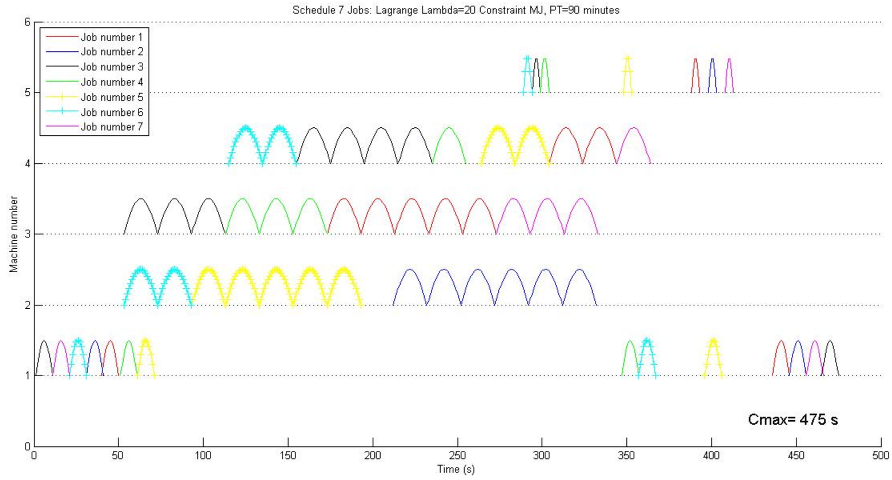


Figure 4.9: Schedule 7 jobs BELTAIP Lagrangian Relaxation MJ constraint, lambda=20

Schedules of Lagrangian relaxation of Ci constraints - 7 jobs BELTAIP - several parameters λ.

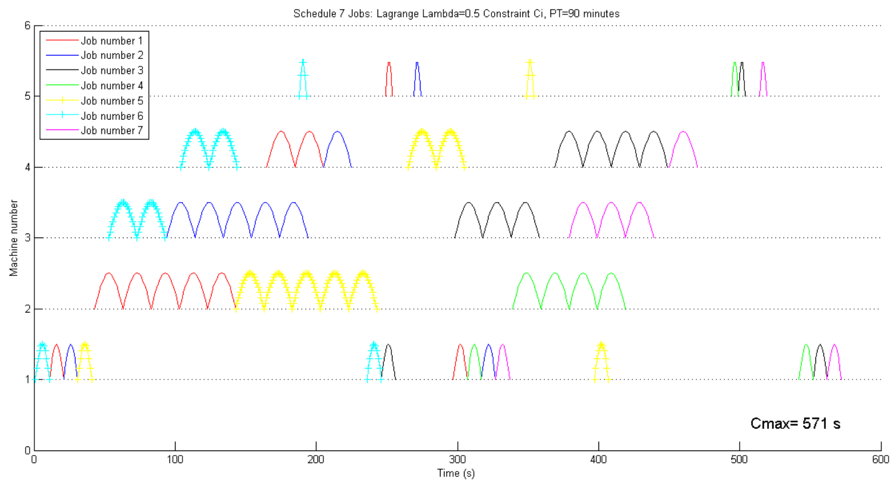


Figure 4.10: Schedule 7 jobs BELTAIP Lagrangian Relaxation Ci constraint, lambda=0.5

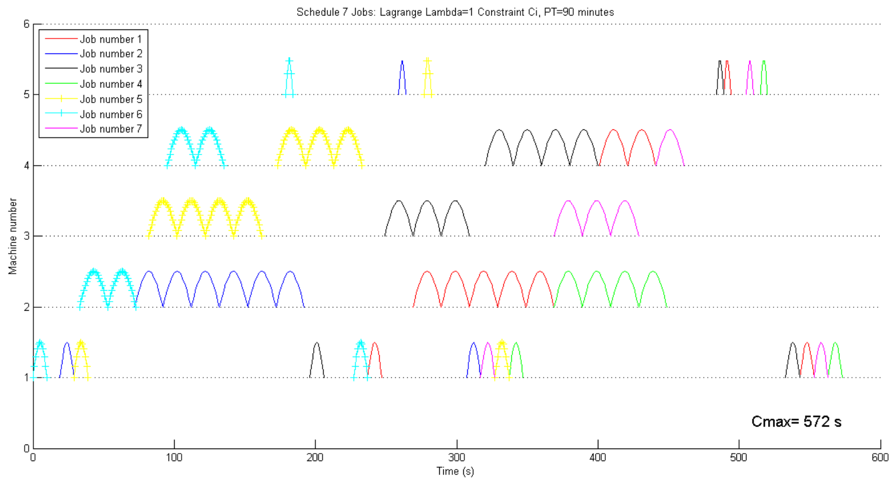


Figure 4.11: Schedule 7 jobs BELTAIP Lagrangian Relaxation Ci constraint, lambda=1

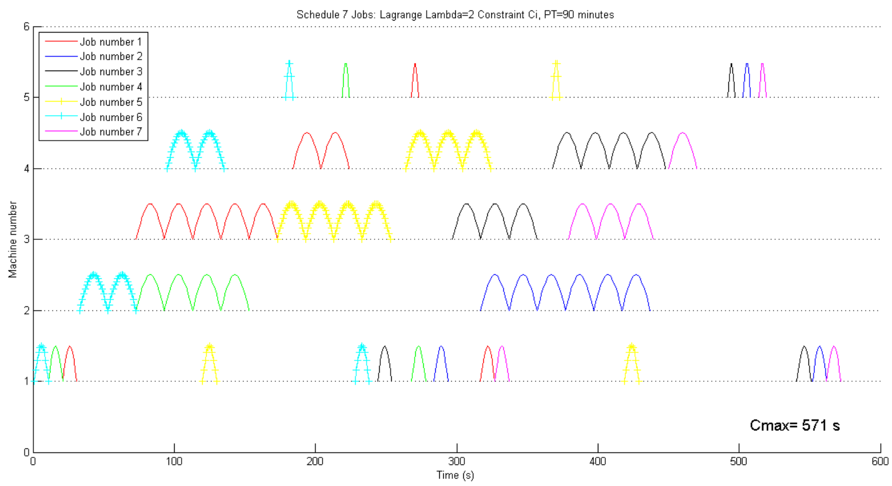


Figure 4.12: Schedule 7 jobs BELTAIP Lagrangian Relaxation Ci constraint, lambda=2

4.2.2.2 Experiment 2 (smaller transportation times)

The Table 4.5 and 4.6 resume the optimization conducted with Lagrangian relaxation of the "MJ" constraint, (longer transportation times).

For this experiment, only the MJ constraints Lagrangian relaxation was taken in count. The Ci constraint relaxation was ruled out for not having acceptable performance. Also, there were chosen two λ values from the previous experiment $\lambda = [0.5, ..., 0.5]$ and $\lambda = [1, ..., 1]$.

The values presented on the Tables 4.5 and 4.6 generate several the schedules depending on the applied parameter λ and the applied parameter λ.

Scenario	Cmax (s)	Optimal/ CPUTime (s)	Comment
3 jobs AIP	219	Yes/3.90	Same as default
4 jobs BELT	276	Yes/27.8	Same as default
6 jobs 2AIP	309	No/1798.54	Better than default - Gap 3.24%
8 jobs 2BELT	446	No/1795.26	Same as default - Gap 27.35%

Table 4.5: *Cmax* for different scenarios, Lagrangian relxation MJ. lambda =0.5

Scenario	Cmax (s)	Optimal/ CPUTime (s)	Comment
3 jobs AIP	219	Yes/2.71	Same as default
4 jobs BELT	276	Yes/17.43	Same as default
6 jobs 2AIP	309	Yes/1258.41	better than default
8 jobs 2BELT	456	No/5394.61	Gap of 20.59%

Table 4.6: *Cmax* for different scenarios, Lagrangian relxation MJ. lambda =1

Schedules of Lagrangian relaxation of MJ constraints - Several scenarios - $\lambda = [0.5, ..., 0.5]$.

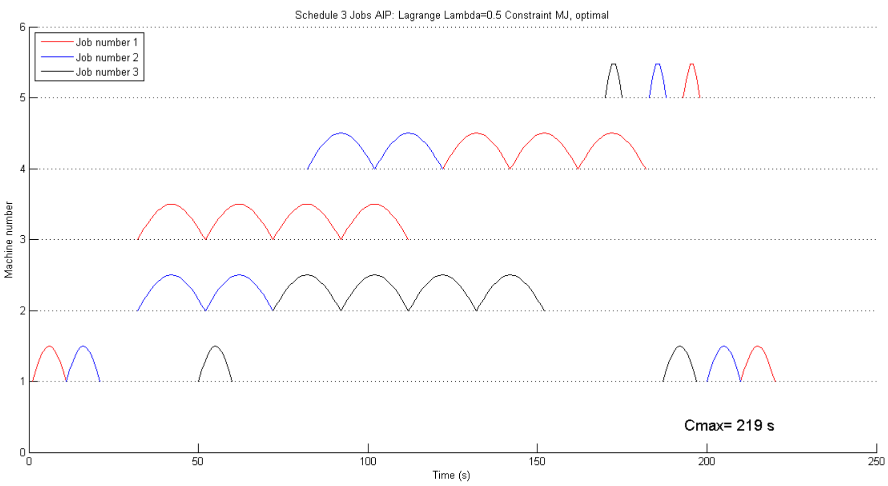


Figure 4.13: Schedule 3 jobs AIP Lagrangian Relaxation MJ constraint, lambda=0.5

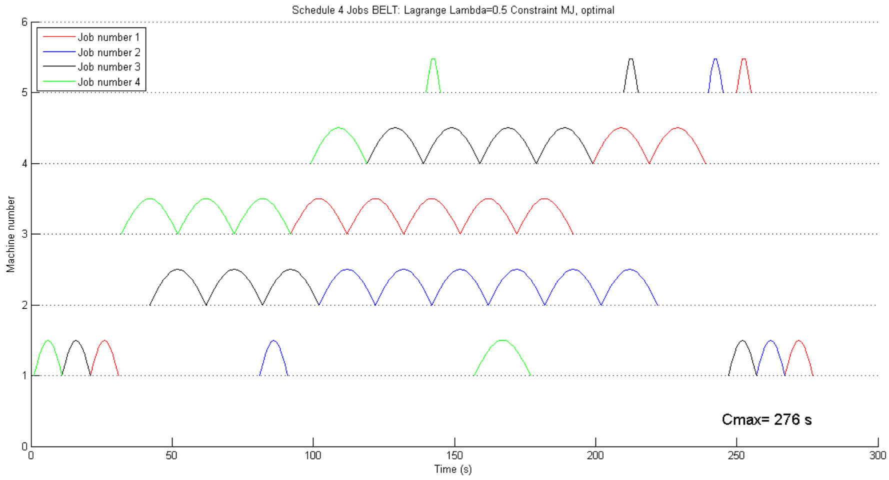


Figure 4.14: Schedule 4 jobs BELT Lagrangian Relaxation MJ constraint, lambda=0.5

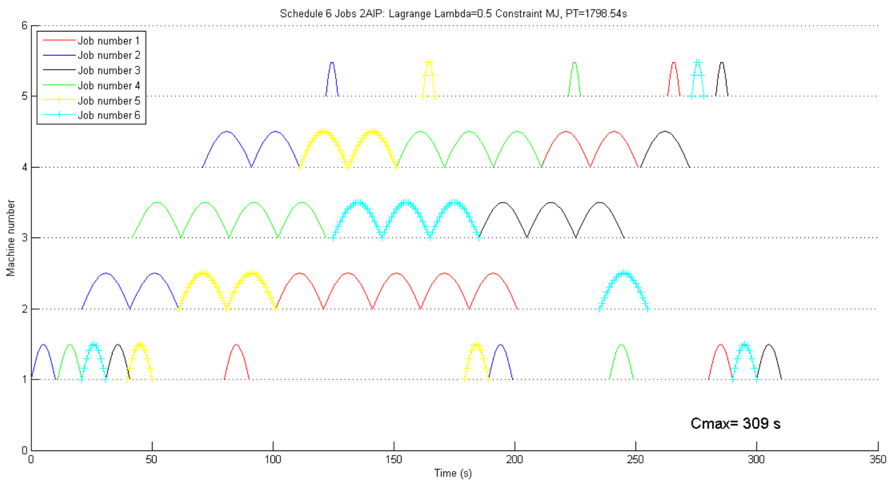


Figure 4.15: Schedule 6 jobs 2AIP Lagrangian Relaxation MJ constraint, lambda=0.5

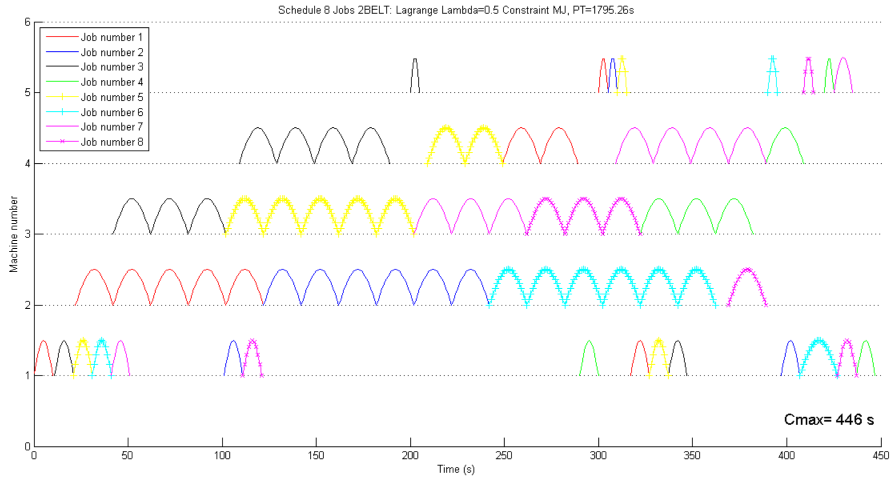


Figure 4.16: Schedule 8 jobs 2BELT Lagrangian Relaxation MJ constraint, lambda=0.5

Schedules of Lagrangian relaxation of MJ constraints - Several scenarios - $\lambda = [1, ..., 1]$.

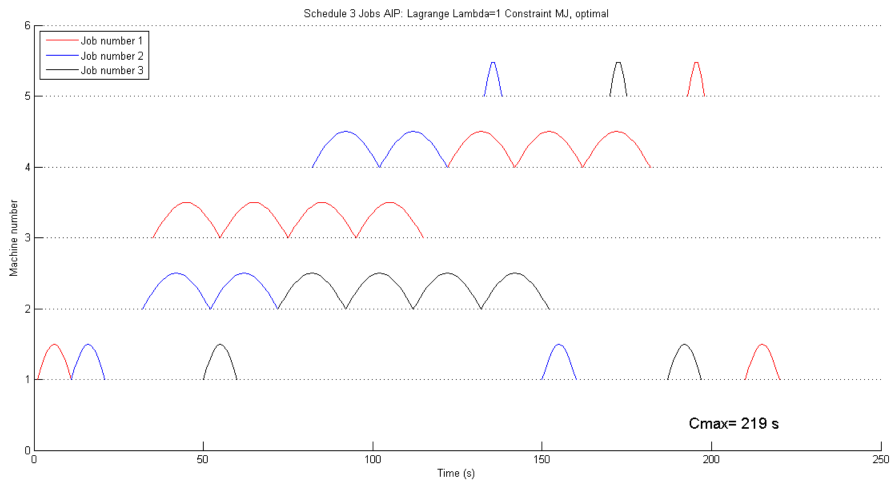


Figure 4.17: Schedule 3 jobs AIP Lagrangian Relaxation MJ constraint, lambda=1

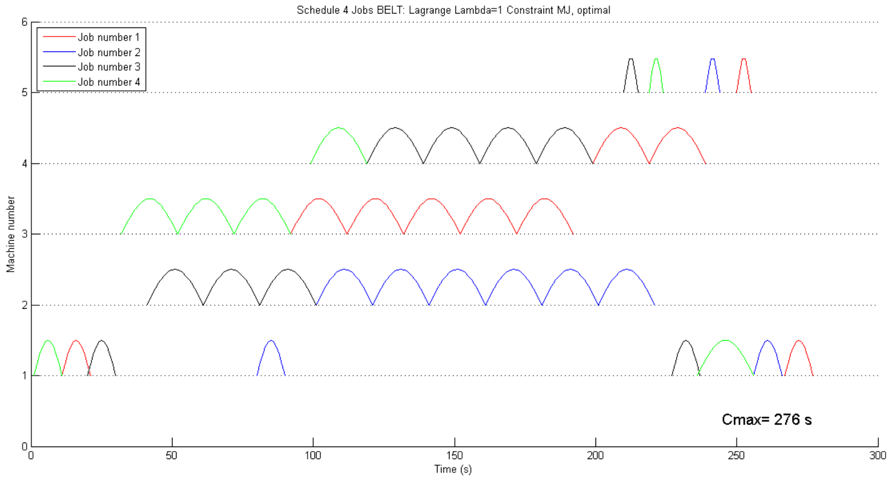


Figure 4.18: Schedule 4 jobs BELT Lagrangian Relaxation MJ constraint, lambda=1

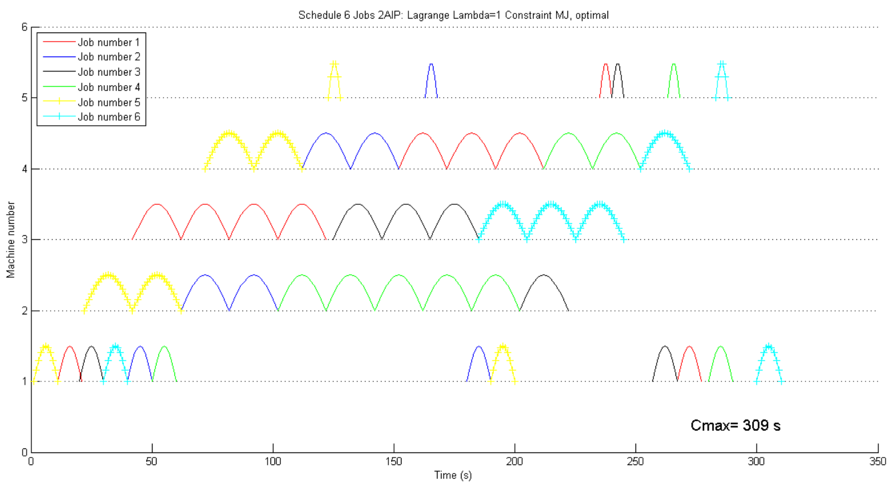


Figure 4.19: Schedule 6 jobs 2AIP Lagrangian Relaxation MJ constraint, lambda=1

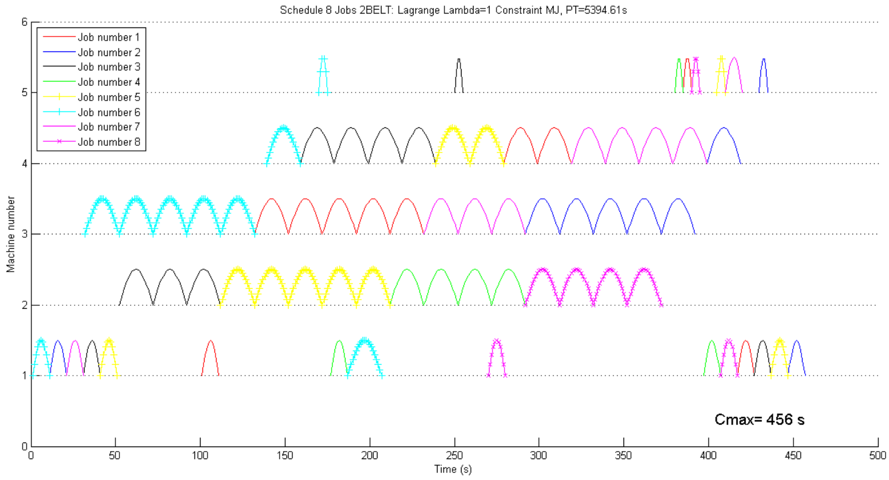


Figure 4.20: Schedule 8 jobs 2BELT Lagrangian Relaxation MJ constraint, lambda=1

Chapter 5

Analysis of the results

The Experiment 1 (7 jobs BELTAIP - Bigger transportation times) helped to test the Lagrangian relaxation with the two constraints MJ and Ci. It also served to obtain a λ parameter that will work adequately regarding the Default case for finding solutions to the scheduling problem.

Looking at the Table 4.3 and Figure 4.2, the solutions to the Lagrangian-relaxed MJ-constraint MILP model always had lower completion times (Cmax) than the Default unrelaxed solutions. Even when the algorithm could not prove to have an optimal solution to the problem, the Lagrangian-relaxed solutions were better than the default case. No matter the value of the λ parameter, the simulations of CPU processing time of 300 s was better than the optimal solution found by CPLEX on 5400 s for the default case. This is a very desirable characteristic for this complex optimization problem.

The Figures 4.4, 4.5, 4.6, 4.7, 4.8 and 4.9 show the production schedules of each solution of the Lagrangian-relaxed MJ-constraint MILP model. Notice that the schedules are feasible, every operation lasts its due time and no operations are interfearing with any other.

The best λvalue is consider to be $\lambda = [0.5, ..., 0.5]$ since it gave the best optimization performance in the shortest time (See Table 4.3).

The solutions from Lagrangian relaxation of Ci constraint are detailed in Table 4.4 and Figure 4.3. Notice that this Lagrangian relaxation did not improve neither the CPU processing time, nor the completion time (Cmax) in almost all of the cases.

The Figures 4.10, 4.11 and 4.12 show the resulting production schedule of each solution of the Lagrangian-relaxed Ci-constraint MILP model. Althought this schedules are all feasible, their completion times Cmax (which is the optimization objective function) are not lower than the Default case. The Lagrangian relaxation of Ci constraints was not applied for the Experiment 2 due to its poor performance.

The Experiment 2 (different production scenarios - smaller transportation times) compiles the TEMPO-PSI team's proposed scenarios on which their interest to improve its optimization results were higher.

The Table 4.2 resumes the Default optimization results of the proposed production scenarios. This values are compared with the Tables 4.5 and 4.6. The solutions obtained on the Experiment 2 were conducted relaxing the MJ constraints for the two chosen λ values, the best found ($\lambda = [0.5, ..., 0.5]$) and an acceptable value ($\lambda = [1, ..., 1]$) for experimentation purposes.

The Lagrangian relaxation of MJ constraints for the Experiment 2 presented better completion times Cmax for the scenario of 6 jobs 2AIP, being the Default case 332 s and the Lagrangian-relaxed optimization results (for both applied λ values) 309 s. It is reelevant to notice that both Lagrangian-relaxed schedules present different combinations on the organization of the operations, althought they have the same completion time. This is expectable since the results from the MILP model are consider as lower bounds.

The scenarios of 3 jobs-AIP and 4 jobs-BELT resulted on the same compeltion times with very low CPU processing times (less than 30 s).

The 8 jobs-2BELT scenario presented the same results as the Default case when the best value of λ was used ($\lambda = [0.5, ..., 0.5]$).

The Figures 4.13, 4.14, 4.15 and 4.16 show the resulting schedules from the Lagrangian relaxation of MJ constraints using $\lambda = [0.5, ..., 0.5]$. The Figures 4.17, 4.18, 4.19 and 4.20 show the resulting schedules from the Lagrangian relaxation of MJ constraints using $\lambda = [1, ..., 1]$. Notice that all the production schedules are feasible and any operation interfiers with other at any time on the same machine.

Chapter 6

Conclusions and future work

The Lagrangian relaxation optimization method was applied to a MILP model representing a FMS. The results proved shorter CPU processing times when the transportation times were bigger.

The Experiment 1 revealed that the MILP model can be relaxed obtaining positive results. The CPU processing times of the lagrange-relaxed MILP model were significantly lower than the Default case.

The Experiment 2 produced similar results to the Default case and in some cases they proved to be better. This was noticed in the reduction of the objective value Cmax (completion time of last operation processed by the Flexible system).

Since the Results from Experiment 2 were not as good as expected (this given by the fact that Experiment 1 presented promising results), the hypothesis for this behavior lies on the fact that Experiment 2 had smaller transportation times, and this condition could affect the performance of the Lagrange-relaxed optimization method.

A deep study on the mathematical model of the AIP-CELL developed by TEMPO-PSI team was conducted with the leading of the TEMPO-PSI team professor in charge, Abdelghani Bekrar, PhD. Some minor modifications were introduced to the formulation due to this study, in order to improve its formality.

Since the AIP-CELL represents a flexible manufacturing system (FMS), other modeling approaches were studied in order to improve the current model. However, most approaches found in literature were destined to Job-Shop Problem (JSP), which made them difficult to adapted to the AIP-Cell FMS case. Graph theory for Flexible Job Shop Problem (FJSP) was study briefly as an alternative approach but no other studies on this specific topic were found so far (only JSP mixed with metaheuristic techniques).

For further research, it is recomended to find the vector λ in the lagrangian relaxation technique through iterative methods, this is practical only when the optimization algorithms prove to have acceptable and feasible solutions for short computer processing times. Iterative methods need to run the optimization algorithm several times in order to find optimal values for λ.

Iterative algorithms for finding an optimal value of the λ vector were found in the literature, some of them are based on penalty methods, subgradient methods and more (See [16] for more on this topic). The Volume algorithm proposed by Barahona & Abnil [1] is a promising method for this objective.

Bibliography

[1] Barahona, F.; Anbil, R. "The volume algorithm: Producing primal solutions with a subgradient method", Mathematical Programming 87 (2000), 385-399.

[2] Bekrar, A.; Zbib, N.; Pach, C.; Sallez, Y.; Trentesaux, D. "An Effective Potential Field Approach to FMS Holonic Heterarchical Control", Elsevier Editorial, Control Engineering Practice, CONENGPRAC-D-11-00341.

[3] Bousbia, S.; Anli, A.; Trentesaux, D.; Grislin, E. "Agile scheduling of flexible manufacturing systems of production", Seventeenth International Association for Mathematics and Computer in Simulation World Congress-IMACS, 2005.

[4] Caumond, A.; Lacomme, P.; Moukrim, A.; Tchernev, N. "An MILP for scheduling problems in an FMS with one vehicle", Elsevier Editorial, European Journal of Operational Research 199 (2009) 706–722, 2008.

[5] Conway, R.W.; Maxwell, W.L.; and Miller, L.W. "Theory of Scheduling", Addison Wesley, Reading, Massachusetts, USA. 1967.

[6] Feitelson, D.; Rudolph, L. "Parallel Job Scheduling: Issues and approaches", Lecture Notes in Computer Science, Springer Berlin / Heidelberg, ISBN: 978-3-540-60153-1 1995.

[7] Gay, D. "IBM ILOG CPLEX V12.1, User's Manual for CPLEX", IBM, 2009.

[8] Gupta, Y. P. and Goyal, S. "Flexibility of manufacturing systems: concepts and measurement", European Journal of Operational Research, 43(2), pp.119-135. 1989.

[9] Heylighen, F. "Combinational Explosion", Web Dictionary of Cybernetics and Systems, <http://cleamc11.vub.ac.be/ASC/COMBIN_EXPLO.html>.

[10] Kuka Robotics, <http://www.kuka-robotics.com/en/>.

[11] Montech Technology, 2010. <http://www.montech.com>.

[12] Nampring, N.; Punglae, S. "The future of Industrial automation flexible manufacturing systems (FMS)", TIM 690 Seminar in Technology and Innovation Management, King Mongkut's University of Technology Thonburi, 2005.

[13] Neumaier, A.; Shcherbina, O. “Safe bounds in linear and mixed-integer linear programming”, MSC primary 90C11, 2003.

[14] Ouelhadj, Djamila and Petrovic, S. “Survey of dynamic scheduling in manufacturing systems. Journal of Scheduling”, 12 (4). pp. 417-431. ISSN 1094-6136, 2009.

[15] Parra, J. “Redes de petri y algoritmos genéticos, una alternativa híbrida para la programación de sistemas de manufactura flexibles” Universidad Católica de Maule, Chile, 2007.

[16] Rocha, A. , "Algoritmos rápidos e estáveis baseados na relaxação Lagrangeana", Tese de doutoramento, Universidade do Minho, 2004.

[17] Sallez, Y.; Berger, T.; Raileanu, S.; Chaabane, S.; Trentesaux, D. “Semi-heterarchical control of FMS: From theory to application”, (2010) Engineering Applications of Artificial Intelligence, 23 (8), pp. 1314-1326.

[18] Stecke, K.; Browne, J. “Variations in flexible manufacturing systems according to the relevant types of automated materials handling”, working paper No. 392, University of Michingan, 1984.

[19] Swamidass, P.M. “Manufacturing flexibility.” In Swamidass, P.M. (ed.) Encyclopaedia of Production and Manufacturing Management. Boston: Kluwer, pp.399-412, 2000.

[20] Trentesaux, D.; Pach, C.; Bekrar, A.; Sallez, Y.; Berger, T.; Leitão, P.; Barbosa, J.; “Benchmarking Framework and Data For Flexible Job Shop Scheduling and Control Problem: part I, the benchmark”, submited to Control Engineering Practice, 2012.

[21] Wago system, <http://www.wago.com>.

Appendices

Appendix A: Matlab code for generating manufacturing scheduling graphics.

```
% Generate Manufacturing Schudule from results.txt
% Victor H. Granados June 2012

clear all
clc

%% Include the data from results.txt into variable sol (solution)
sol=[];
im_title=('prueba'); % Title of the image.

%% Schedule generator

tmax=max(sol(:,6));
jmax=max(sol(:,1));
figure1=figure;
Create axes axes1 = axes('Parent',figure1,'YGrid','on');

% Uncomment the following line to preserve the Y-limits of the axes
ylim(axes1,[0 6]); %Y axis
hold(axes1,'all');

for h=1:(size(sol,1))
    x=(sol(h,5):1:sol(h,6));
    y=sol(h,4)+(1/2)*sin(pi*(x-x(1))/(size(x,2)-1));
    switch sol(h,1) % Column of Job number
        case 1, J1=plot(x,y,'-r');  %Job:1 Color:Red
        case 2, J2=plot(x,y,'-b');  %Job:2 Color:Blue
        case 3, J3=plot(x,y,'-k');  %Job:3 Color:Black
        case 4, J4=plot(x,y,'-g');  %Job:4 Color:Green
        case 5, J5=plot(x,y,'-y+'); %Job:5 Color:Yellow
        case 6, J6=plot(x,y,'-c+'); %Job:6 Color:Cyan
        case 7, J7=plot(x,y,'-m');  %Job:7 Color:Magenta
    end
end

% Image Title
title(im_title);

% Legend on the image
hold off
legend([J1,J2,J3,J4,J5,J6,J7], 'Location', 'NorthWest', 'Job number 1'
    , 'Job number 2', 'Job number 3', 'Job number 4', 'Job number 5', '
    Job number 6', 'Job number 7')
```

```
% Text with Cmax value plotted in figure
index = find(max(sol(:,6)) == sol(:,6)); % Find the index of the Cmax

text(sol(index,6),0.5,['Cmax= ',num2str(tmax),' s'], 'VerticalAlignment', 'top', 'HorizontalAlignment', 'right', 'FontSize', 14)

% Create xlabel
xlabel('Time (s)');

% Create ylabel
ylabel('Machine number');

%%%%% End of program %%%%%
```

Appendix B: The Volume algorithm proposed by Barahona & Abnil [1].

Input: π^0

Initialization: Get y^0, an optimal solution in $z(\pi^0)$ and define $v^0 = b - Ay^0 \in \partial z(\pi^0)$.

Start $\bar{\pi}^1 = \pi^0$, $x^1 = y^0$, $w^1 = v^0$, $j = 1$ and $k = 1$.

Generic Iteration in j:

Step 1: For some $s_j > 0$, define $\pi^j = [\bar{\pi}^k + s_j w^j]^+$.

Step 2: Get y^j, an optimal solution in $z(\pi^j)$ and define $v^j = b - Ay^j \in \partial z(\pi^j)$.

Step 3: For some $\alpha_j \in [0,1]$, define

$$x^{j+1} = \alpha_j\, y^j + (1 - \alpha_j)x^j$$

$$w^{j+1} = \alpha_j\, v^j + (1 - \alpha_j)w^j$$

Step 4: If $z(\pi^j) > z(\bar{\pi}^k)$ then

Define $\bar{\pi}^{k+1} = \pi^j$ and do $k \leftarrow k + 1$

Step 5: Test the stopping criteria. Do $j \leftarrow j + 1$ and go back to step 1.

Figure 6.1: Volumen algorithm proposed by Barahona & Abnil.