

# Novel Approaches to the Formulation of Scheduling Problems

José Manuel García-Sánchez \* and Plácido Moreno 

Department of Industrial Management I, University of Seville, E41092 Seville, Spain; placidomb@us.es

\* Correspondence: jmg@us.es

**Abstract:** This paper presents two novel formulations for scheduling problems, namely order-position hybrid formulation (OPH) and order-disjunctive hybrid formulation (ODH), which extend and combine parts of existing formulation strategies. The first strategy (OPH) is based on sequence position and linear ordering formulations, adding relationships between constraints that allow relaxing some decision variables. The second approach (ODH) is based on linear ordering and disjunctive formulations. In this work, we prove ODH to be the most efficient formulation known so far. The experiments have been carried out with a large set of problems, which consider single machines and identical parallel machines. Computational results show that OPH is better than the rest of the existing formulations for the case of weighted completion objectives, while ODH turns out to be the best approach for most scenarios studied.

**Keywords:** machine scheduling; weighted completion problem; weighted tardiness; mixed integer programming

**MSC:** 90B35; 90C11

## 1. Introduction

In this paper, two new approaches to formulate machine scheduling problems are proposed. These new formulations extend and mix parts of previous models. We focus on two scenarios, namely a single machine and identical parallel machines, in order to be able to study a wide range of problems. The set of problems to be considered is  $1 || L_{MAX}$ ,  $1 || \sum U_j$ ,  $1 || \sum w_j C_j$ ,  $1 || \sum T_i$ ,  $1 || \sum w_i T_i$ ,  $P_m || \sum w_j C_j$ , and  $P_m || \sum w_i T_i$ .

From a general point of view, the main formulations for single-machine scheduling have been the following:

- DC: disjunctive constraints formulation [1];
- LO: linear ordering formulation [2,3];
- TI: time-indexed formulation [4];
- SP: sequence position formulation [5,6].

An experimental study of the previous formulations on a single machine was carried out [7] by considering objectives such as  $1 || \sum w_j C_j$ ,  $1 || L_{MAX}$ ,  $1 || \sum w_j T_j$ , and  $1 || \sum U_j$ , as well as the same problems with release dates. Another experimental study of these formulations on the single-machine total tardiness problem was performed in [8], including a hybrid formulation between the LO formulation and the SP formulation, with an adaptation of the traveling salesman problem formulation with Desrochers and Laporte anti-loop constraints [9]. Although these last two approaches are more recent, neither of them achieves any relevant improvements over the other formulations. The DC and SP formulations were also compared empirically in other single-machine scenarios [10].

However, as far as we are concerned, there has not been a relevant contribution to the formulation of scheduling problems for years. Therefore, the main novelty of this paper is to develop new strategies for modeling scheduling problems, aimed at both single-machine and parallel-machine scenarios.



**Citation:** García-Sánchez, J.M.; Moreno, P. Novel Approaches to the Formulation of Scheduling Problems. *Mathematics* **2024**, *12*, 1035. <https://doi.org/10.3390/math12071035>

Academic Editor: Amir Elalouf

Received: 4 March 2024

Revised: 21 March 2024

Accepted: 27 March 2024

Published: 30 March 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Considering the results from previous studies and the results presented in this work, it can be concluded that the TI formulation is the best formulation for short job sizes, such as the uniform distribution from the interval [1, 10]. However, with the increase in job size, it becomes a very inefficient formulation. For instance, the time to build the model in a single-machine problem with no more than 100 jobs can take several hours if the job size is in the uniform distribution from the interval [1, 100].

When the problem objective does not imply making the job completion time explicit, SP is the best formulation for any job size [8], but the objective of the problem can be modeled using the completion time of each position, which happens in problems like  $1||\Sigma C_j$ ,  $1||\Sigma T_j$ ,  $1||L_{MAX}$ , or  $1||\Sigma U_j$ . However, weighted objectives in linear programming cannot be modeled without obtaining the completion time for each job. For these cases, the SP formulation loses efficiency, although its behavior depends on the specific weighted objective. Regarding weighted completion times, its behavior becomes very inefficient, whereas its results improve for weighted tardiness purposes.

LO is a good formulation for problems with few jobs [7]. The set of  $\theta(n^3)$  transitivity constraints, where  $n$  refers to the number of jobs, significantly worsens its performance when the number of jobs increases. Finally, DC is the formulation with the worst behavior, even for problems with the computational complexity of class P [8].

The next section shows the modeling of these formulations for single-machine problems. In Section 3, we present the new approaches to formulate single-machine problems. In Section 4, we adapt these formulations for a parallel-machine scenario. Finally, in Section 5, an empirical analysis of the formulations is developed.

## 2. Overview of Previous Formulations

In this section, a summary is presented for each of the four to-date best approaches for weighted objectives, namely the DC, LO, TI, and SP formulations.

Let  $N = \{1..n\}$  be the set of jobs to be sequenced.

$\forall j \in N$ ; let  $p_j$  be the processing time,  $w_j$  be the weight of the job, and  $d_j$  be the due date of the job. Also,  $R = \sum_{j=1}^n p_j$ .

### 2.1. Disjunctive Constraints Formulation (DC)

The variables of the formulation are defined as follows:

$\forall j, k \in N / j \neq k : \alpha_{jk} = 1$  if job  $j$  precedes job  $k$ ; 0 otherwise;

$\forall j \in N : C_j =$  completion time of job  $j$ ;  $L_{MAX} =$  maximum lateness;

$\forall j \in N : T_j =$  tardiness time of job  $j$ ;  $\forall j \in N : U_j = 1$  if job  $j$  is late; 0 otherwise.

Common constraints in all DC single-machine models are (1)–(3), i.e., constraints (1)–(3) are included for all objectives:

$$\forall j, k \in N / j \neq k : \alpha_{jk} + \alpha_{kj} = 1, \tag{1}$$

$$\forall j \in N : C_j \geq p_j, \tag{2}$$

$$\forall j, k \in N / j \neq k : C_j + p_k \leq C_k + R(1 - \alpha_{jk}), \tag{3}$$

$$\text{For } 1||L_{MAX} : \forall j \in N : L_{MAX} \geq C_j - d_j, \tag{4}$$

$$\text{Min } L_{MAX}, \tag{5}$$

$$\text{For } 1||U_j : \forall j \in N : C_j - d_j \leq \left(\sum_{j=1}^n p_j - d_j\right)U_j, \tag{6}$$

$$\text{Min } \sum_{j=1}^n U_j, \tag{7}$$

$$\text{For } 1||\Sigma w_j C : \text{Min } \sum_{j=1}^n w_j C_j, \tag{8}$$

$$\text{For } 1||\Sigma T_j : \forall j \in N : T_j \geq C_j - d_j, \tag{9}$$

$$\text{Min } \sum_{j=1}^n T_j, \tag{10}$$

For  $1 \leq j \leq n$ : constraints (9),

$$\text{Min } \sum_{j=1}^n w_j T_j, \tag{11}$$

### 2.2. Linear Ordering Formulation (LO)

It uses the same variables that were defined for the DC formulation. Common constraints in all LO single-machine models are the following:

Precedence constraints : constraints (1),

Calculation of lower bound for the time completion:

$$\forall j \in N : C_j \geq \sum_{\substack{k \in N \\ k \neq j}} p_k \alpha_{kj} + p_j, \tag{12}$$

Transitivity constraints:

$$\forall j, k, i \in N / j \neq k \neq i : \alpha_{jk} + \alpha_{ki} + \alpha_{ij} \leq 2, \tag{13}$$

All objectives for scheduling problems are modeled in the same way as that for the DC formulation (Section 2). The only difference is the replacement of common constraints (2) and (3) by constraints (12) and (13).

### 2.3. Time-Indexed Formulation (TI)

For job processing, let  $t = 1, 2, \dots, T$  be the set of time periods, where  $T = \sum_{j=1}^n p_j$ .

The variables of the formulation are defined as follows:

$\forall j \in N, t \in \{1..T - p_j + 1\} : x_{jt} = 1$  if job  $j$  starts processing at time  $t$ ; 0 otherwise;

$\forall j \in N : C_j =$  completion time of job  $j$ .

Common constraints in all TI single machine models are:

$$\text{Process each job once : } \forall j \in N : \sum_{t=1}^{T-p_j+1} x_{jt} = 1, \tag{14}$$

No more than one job is processed at each time:

$$t = 1 \dots T : \sum_{j=1}^n \sum_{s=\max(1,t-p_j+1)}^t x_{js} \leq 1, \tag{15}$$

Objective functions for each type of problem are:

$$\text{For } 1 \parallel L_{MAX} : \forall j \in N : C_j = \sum_{t=1}^{T-p_j+1} (t + p_j - 1) x_{jt}, \tag{16}$$

expressions (4) and (5),

$$\text{For } 1 \parallel U_j : \text{Min } \sum_{j=1}^n \sum_{t=d_j-p_j+1}^{T-p_j+1} x_{jt}, \tag{17}$$

$$\text{For } 1 \parallel \sum w_j C : \text{Min } \sum_{j=1}^n \sum_{t=1}^{T-p_j+1} w_j (t + p_j - 1) x_{jt}, \tag{18}$$

$$\text{For } 1 \parallel \sum T_j : \text{Min } \sum_{j=1}^n \sum_{t=1}^{T-p_j+1} \max(0, t + p_j - 1 - d_j) x_{jt}, \tag{19}$$

$$\text{For } 1 \parallel \sum W_j T_j : \text{Min } \sum_{j=1}^n \sum_{t=1}^{T-p_j+1} w_j \max(0, t + p_j - 1 - d_j) x_{jt}, \tag{20}$$

### 2.4. Sequence Position Formulation (SP)

Let  $S = \{1..n\}$  be the set of positions in the sequence. Since  $n$  is the number of jobs,  $n$  is also the number of positions in the sequence.

The variables of the SP formulation are defined as follows:

$\forall j \in N, s \in S : \beta_{js} = 1$  if job  $j$  is processed at position  $s$ ; 0 otherwise;

$\forall s \in S : C_s^P =$  completion time of position  $s$ ;  
 $\forall s \in S : U_s = 1$  if job in position  $s$  is late, 0 otherwise;  
 $\forall j \in N : C_j =$  completion time of job  $j$ ;  
 $\forall s \in S : T_s^P =$  tardiness time of position  $s$ ;  
 $\forall j \in N : T_j =$  tardiness time of job  $j$ .

Common constraints in all SP single-machine models are:

$$\text{Each job at a position : } \forall j \in N : \sum_{s=1}^n \beta_{js} = 1, \tag{21}$$

$$\text{In each position a job : } \forall s \in S : \sum_{j=1}^n \beta_{js} = 1. \tag{22}$$

Calculation of the lower bound for the completion time of each position:

$$\forall s \in S : C_s^P \geq \sum_{j=1}^n p_j \sum_{k=1}^s \beta_{jk}, \tag{23}$$

$$\text{For } 1||L_{MAX} : \forall s \in S : L_{MAX} \geq C_s^P - \sum_{j=1}^n d_j \beta_{js}, \tag{24}$$

and expression (5).

$$\text{For } 1||U_j : \forall s \in S : C_s^P \leq \sum_{j=1}^n d_j \beta_{js} + RU_s, \tag{25}$$

$$\text{Min } \sum_{s=1}^n U_s. \tag{26}$$

In the previous cases, where no weighted objectives were imposed, there was no need to work with variables considering the job. In other words, the model could include only variables relative to the position. However, in this case, we need to calculate the lower bound for the completion time of each job from the completion time of each position:

$$\text{For } 1||\sum w_j C_{jj} : \forall j \in N, s \in S : C_j \geq C_s^P - R(1 - \beta_{js}), \tag{27}$$

and expression (8).

$$\text{For } 1||\sum T_j : \forall s \in S : T_s^P \geq C_s^P - \sum_{j=1}^n d_j \beta_{js}, \tag{28}$$

$$\text{Min } \sum_{s=1}^n T_s^P. \tag{29}$$

For  $1||\sum w_j T_j$  : expressions (9), (11), and (26).

### 3. Novel Formulations—Single-Machine Scenario

In this section, the formulations of the novel approaches are presented for the case of single-machine problems. The parallel-machine scenario will be developed in Section 4.

#### 3.1. Order-Position Hybrid Formulation (OPH)

This novel approach takes elements from both the LO formulation and SP formulation, so that the transitive constraints can be replaced. It uses the following variables:

$\forall j, k \in N / j \neq k : \alpha_{jk} = 1$  if job  $j$  precedes job  $k$ ; 0 otherwise;  
 $\forall j \in N, s \in S : \beta_{js} = 1$  if job  $j$  is processed at position  $s$ ; 0 otherwise;  
 $\forall j \in N : C_j =$  completion time of job  $j$ ;  
 $\forall j \in N : T_j =$  tardiness time of job  $j$ .

Common constraints in all DC single-machine models are expressions (1) and (12) from the LO formulation and (22) from the SP formulation:

$$\begin{aligned} \forall j, k \in N / j \neq k : \alpha_{jk} + \alpha_{kj} &= 1, \\ \forall j \in N : C_j &\geq \sum_{\substack{k \in N \\ k \neq j}} p_k \alpha_{kj} + p_j, \\ \forall s \in S : \sum_{j=1}^n \beta_{js} &= 1. \end{aligned}$$

To establish a connection between formulations and prevent precedents from forming a loop, there is a need to incorporate the following relationship between both formulations into the model:

$$\forall j \in N : \sum_{\substack{k \in N \\ k \notin j}} \alpha_{kj} + 1 = \sum_{s=1}^n s\beta_{js} \tag{30}$$

Constraint (30) ensures that the position in the sequence occupied by job  $j$  is equal to the number of jobs preceding job  $j$ , plus one. Constraint (30) also forces each job to have a position, thus making it unnecessary to include constraint (21).

**Proof.** Because of constraint (22), there must be a job in each position. If a job were to be in more than one position, some job  $j$  would not be able to be placed in any position; hence,  $\sum_{k=1}^n \beta_{jk} = 0$ . By constraint (30), for that job  $j$ :  $\sum_{\substack{k \in N \\ k \notin j}} \alpha_{kj} = -1$ , which is impossible.

Therefore, each job has a unique integer position in the sequence, which corresponds to the integer  $\sum_{k=1}^n k\beta_{jk}$ ,  $\forall j \in N$ .  $\square$

In addition, the set of constraint (30) allows us to relax variables  $\alpha_{jk}$ .

**Proof.** Let the established order be:

$k = 1$  Job  $o_1$ ;  $k = 2$  Job  $o_2$ ;  $\dots$ ;  $k = i$  Job  $o_i$ ;  $\dots$ ;  $k = n$  Job  $o_n$

There is a bijective mapping between  $\{o_1, o_2, \dots, o_n\}$  and the set of jobs  $\{1..n\}$ .

Position( $o_1$ ) = 1:

$$\begin{aligned} o_1 := \text{by constraints (30)} &\Rightarrow \sum_{\substack{o_k \in N \\ o_k \neq o_1}} \alpha_{o_k o_1} + 1 = 1 \Rightarrow \sum_{\substack{o_k \in N \\ o_k \neq o_1}} \alpha_{o_k o_1} = 0 \\ &\Rightarrow \forall k/k > 1 : \alpha_{o_k o_1} = 0 \Rightarrow \text{by constraints (1)} \Rightarrow \forall k/k > 1 : \alpha_{o_1 o_k} = 1 \end{aligned} \tag{31.a}$$

Position( $o_2$ ) = 2:

$$\begin{aligned} \text{by constraints (30)} &\Rightarrow \sum_{\substack{o_k \in N \\ o_k \neq o_2}} \alpha_{o_k o_2} = 1; \\ \text{by expressions (31.a)} &\Rightarrow \alpha_{o_1 o_2} = 1 \Rightarrow \\ \forall k/o_k \neq o_2 \&\&o_k \neq o_1 (\equiv k > 2) : \alpha_{o_k o_2} = 0 \Rightarrow \\ \text{by constraints (1)} &\Rightarrow \forall k/k > 2 : \alpha_{o_2 o_k} = 1 \end{aligned} \tag{31.b}$$

...

Position( $o_i$ ) =  $i$ :

$$\begin{aligned} \text{by constraints (30)} &\Rightarrow \sum_{\substack{o_k \in N \\ o_k \neq o_i}} \alpha_{o_k o_i} = i - 1; \\ \text{by expressions (31.a)} &\Rightarrow \alpha_{o_1 o_i} = 1 \\ \text{by expressions (31.b)} &\Rightarrow \alpha_{o_2 o_i} = 1 \end{aligned} \tag{31.i}$$

...

$$\begin{aligned} \text{by expressions (31.i - 1)} &\Rightarrow \alpha_{o_{i-1} o_i} = 1 \Rightarrow \\ \forall k/k > i : \alpha_{o_k o_i} = 0 &\Rightarrow \text{by constraints (1)} \Rightarrow \forall k/k > i : \alpha_{o_i o_k} = 1 \end{aligned}$$

...

Position( $o_n$ ) = n:

$$\begin{aligned}
 & \text{by constraints (30)} \Rightarrow \sum_{\substack{o_k \in N \\ o_k \neq o_n}} \alpha_{o_k o_n} = n - 1; \\
 & \text{by expressions (31.a)} \Rightarrow \alpha_{o_1 o_n} = 1 \\
 & \text{by expressions (31.b)} \Rightarrow \alpha_{o_2 o_n} = 1 \\
 & \dots \\
 & \text{by expressions (31.i)} \Rightarrow \alpha_{o_i o_n} = 1 \\
 & \dots \\
 & \text{by expressions (31.n - 1)} \Rightarrow \alpha_{o_{n-1} o_n} = 1 \Rightarrow
 \end{aligned} \tag{31.n}$$

Therefore, all variables  $\alpha_{jk}$  are assigned a value of 1 or 0, even if they are defined as continuous.  $\square$

For all objectives, the constraints and expressions to be included correspond to those proposed for the DC or LO formulations.

### 3.2. Order-Disjunctive Hybrid Formulation (ODH)

The LO formulation problem shows a high number of  $O(n^3)$  of transitive constraints (constraint (13)), which prevents precedence loops amongst jobs. We propose to replace transitive constraints with the following disjunctive constraint (32), which only employs  $n^2$  constraints while guaranteeing the proper sequence of the jobs:

$$\forall j, k \in N / j \neq k : C_j + p_k \alpha_{jk} \leq C_k + R \alpha_{kj}, \tag{32}$$

which are equivalent to constraint (3).

Therefore, common constraints of the ODH formulation will be expressions (1) and (12). The other expressions correspond to those of the DC or LO formulation.

## 4. Formulations for Parallel-Machine Scenario

In this scenario, we need to incorporate set  $M=\{1..m\}$  of parallel machines as data. Precedence relationships for LO, OPH, and ODH formulations must only happen amongst jobs processed on the same machine. The machine where each job is processed needs to be considered a decision variable for the LO formulation, but not for the OPH formulation, where it can be treated as an auxiliary calculation of the position in the sequence of each machine, as in the SP formulation. The DC formulation is not going to be studied in this section because of its poor behavior. Finally, we are going to focus on the problems with objectives  $P_m \mid \mid \sum w_j C_j$  and  $P_m \mid \mid \sum w_j T_j$ . Other objectives for the study of identical parallel machines were considered in [11–13].

### 4.1. Linear Ordering Formulation (LO)

The variables of the formulation are defined as follows:

$\forall j, k \in N / j \neq k : \alpha_{jk} = 1$  if job  $j$  precedes job  $k$ ; 0 otherwise;

$\forall j \in N, i \in M : \omega_{ji} = 1$  if job  $j$  is processed in machine  $i$ ; 0 otherwise;

$\forall j, k \in N / j < k : \delta_{jk} = 1$  if jobs  $j$  and  $k$  are processed in the same machine; 0 otherwise;

$\forall j \in N : C_j =$  completion time of job  $j$ ;

$\forall j \in N : T_j =$  Tardiness time of job  $j$ .

Common constraints in all LO parallel-machine models are:

$$\text{Precedence constraints : } \forall j, k \in N / j \neq k : \alpha_{jk} + \alpha_{kj} = \delta_{jk}, \tag{33}$$

$$\text{Activation of } \delta_{jk}: \forall i \in M, \forall j, k \in N / j < k : \omega_{ji} + \omega_{ki} \leq 1 + \delta_{jk}, \tag{34}$$

$$\text{A machine of each job : } \forall j \in N : \sum_{i=1}^m \omega_{ji} = 1. \tag{35}$$

In addition, constraints (12) and (13) need to be incorporated. The objective functions for every problem are the same as those of the single-machine scenario.

4.2. Time-Indexed Formulation (TI)

For job processing, let  $t = 1, 2, \dots, T$  be the set of time periods, where  $T = \sum_{j=1}^n p_j$ .

The variables of the formulation are defined as follows:

$\forall i \in M, \forall j \in N, t \in \{1..T - p_j + 1\} : x_{jti} = 1;$

If job  $j$  starts processing at time  $t$  in machine  $i$ ; 0 otherwise;

$\forall j \in N : C_j =$  completion time of job  $j$ .

Common constraints in all TI parallel-machine models are:

$$\text{Process each job once : } \forall j \in N : \sum_{i=1}^m \sum_{t=1}^{T-p_j+1} x_{jti} = 1. \tag{36}$$

No more than one job is processed at each time:

$$\forall i \in M, t = 1 \dots T : \sum_{j=1}^n \sum_{s=\max(1,t-p_j+1)}^t x_{jsi} \leq 1. \tag{37}$$

Objective functions for each type of problem are:

$$\text{For } 1||\sum w_j C : \text{ Min } \sum_{i=1}^m \sum_{j=1}^n \sum_{t=1}^{T-p_j+1} w_j (t + p_j - 1) x_{jti}, \tag{38}$$

$$\text{For } 1||\sum w_j T_j : \text{ Min } \sum_{i=1}^m \sum_{j=1}^n \sum_{t=1}^{T-p_j+1} w_j \max(0, t + p_j - 1 - d_j) x_{jti}. \tag{39}$$

4.3. Sequence Position Formulation (SP)

The variables of the SP formulation are defined as follows:

$\forall j \in N, s \in S, i \in M : \beta_{jsi} = 1$  if job  $j$  is processed at position  $s$  in machine  $i$ ; 0 otherwise;

$\forall s \in S : C_s^P =$  completion time of position  $s$ ;

$\forall j \in N : C_j =$  completion time of job  $j$ ;

$\forall j \in N : T_j =$  tardiness time of job  $j$ .

Common constraints in all SP parallel-machine models are:

$$\text{Each job at a position : } \forall j \in N : \sum_{s=1}^n \sum_{i=1}^m \beta_{jsi} = 1. \tag{40}$$

No more than one job in each position:

$$\forall s \in S, \forall i \in M, : \sum_{j=1}^n \beta_{jsi} \leq 1. \tag{41}$$

Calculation of lower bound for the time completion of each position:

$$\forall s \in S, \forall i \in M : C_{si}^P \geq \sum_{j=1}^n p_j \sum_{k=1}^s \beta_{jki}. \tag{42}$$

Calculation of lower bound for the time completion of each job:

$$\forall s \in S, \forall i \in M : C_{si}^P \geq \sum_{j=1}^n p_j \sum_{k=1}^s \beta_{jki}. \tag{43}$$

The objective functions for each type of problem are the same as those defined in the LO formulation.

4.4. Order-Position Hybrid Formulation (OPH)

All variables from the LO formulation and variables  $\beta_{jsi}$  from the SP formulation are used in the OPH formulation.

Expressions (33) and (34) from the LO formulation and (40) and (41) from the SP formulation need to be incorporated. The relationship between the LO and SP formulations is defined as:

$$\forall s \in S, \forall i \in M, : \sum_{j=1}^n \beta_{jsi} \leq 1. \tag{44}$$

As in expression (30), variables  $\alpha_{jk}$  can be considered continuous. Regarding variables  $\delta_{jk}$ , their value is set to 0 whenever any variable  $\omega_{ji}$  or  $\omega_{jk}$  is equal to 0. Therefore, the next constraints must be added:

$$\forall i \in M, \forall j, k \in N / j < k : \omega_{ji} - \omega_{ki} \geq \delta_{jk} - 1, \tag{45}$$

$$\forall i \in M, \forall j, k \in N / j < k : \omega_{ki} - \omega_{ji} \geq \delta_{jk} - 1. \tag{46}$$

However, variables  $\omega_{ji}$  are not decision variables in this model but are calculated from variables  $\beta_{jsi}$  by using the following expression:

$$\forall j \in N, \forall i \in M : \omega_{ji} = \sum_{s=1}^n \beta_{jsi} \tag{47}$$

The objective functions for each type of problem are the same as those defined in the LO formulation.

#### 4.5. Order-Disjunctive Hybrid Formulation (ODH)

The ODH formulation with parallel machines uses variables from the LO formulation, along with its constraints, (33)–(35) and (12). In addition, constraint (3) must be added to guarantee proper sequences on each machine. Constraint (32), which was defined for the ODH formulation in a single machine, cannot be applied in the case of parallel machines, since two jobs on different machines do not have precedence relationships.

The objective functions for each type of problem are the same as those defined in the LO formulation.

### 5. Computational Results

Experiments have been performed by following the experimental analysis proposed by [14,15]. Therefore, the following parameters have been considered:

- Processing time  $p_j$ : generated from two uniform distributions: [1, 100] and [1, 10];
- Weight  $w_j$ : generated from the uniform distribution [1, 10];
- Due date  $d_j$ : an integer generated from the uniform distribution  $[P(L - R/2), P(L + R/2)]$ , where P is the sum of the processing times of all jobs. Parameters L and R correspond to relative measures of the location and range of the distribution, respectively, and take three values each: {0.2, 0.4, 0.6} and {0.6, 1, 1.4};
- Number of jobs ( $n$ ): six sizes have been considered {20, 40, 60, 80, 100, and 200}, although no problem includes the six sizes.

From each combination of parameters, four instances are generated. Experiments were run on an Intel(R) Core(TM) i7-10700K CPU @ 3.80 GH with 16 Gb RAM. The optimization library GUROBI Optimizer 10.0.1 was used.

All instances for all problems will be executed during a number of seconds equal to the number of jobs times the number of machines in the corresponding problem, which is a significant difference compared to the analysis of previous scheduling models, where the execution time is usually one hour.

#### 5.1. Linear Programming Relaxation

The linear programming relaxation has been analyzed for  $1 \parallel \sum w_j C_j$  and  $1 \parallel \sum w_j T_j$ . This relaxation of the integer models produces optimal solutions for the formulations TI, LO, OPH, and ODH in the case of problem  $1 \parallel \sum w_j C_j$ . However, the SP formulation obtains solutions with an integrality gap of 97.5%.

Regarding problem  $1 \mid \mid \Sigma w_j T_j$ , Table 1 shows the lower bound gap. In this case, the TI formulation presents the best bounds. ODH also shows a better gap than the rest of the formulations. LO and OPH have the same lower bounds.

**Table 1.** Lower Bound GAP (%).

	LO	TI	SP	OPH	ODH
GAP (%)	55.9	1.5	65.4	55.9	18.5

5.2. Results for  $1 \mid \mid \Sigma w_j C_j$

Regarding the computational results of the integer resolution of the problem, Table 2 shows the average resolution times for the four instances of each size and  $p_i$  interval. The behaviors of the LO, TI, OPH, and ODH models can be observed. However, results from the sequencing-position model (SP) were not included because, with just 20 jobs and after an hour, no instance had finished, and the best solution found so far was not optimal.

**Table 2.** Summary of computational results on problem  $1 \mid \mid \Sigma w_j C_j$ .

	N	LO	TI	OPH	ODH
$p_j \in [1, 100]$	20	0.02	3.21	0.02	0.02
	40	0.18	17.62	0.04	0.11
	60	0.78	39.62	0.09	0.21
	100	4.31	115.64	0.26	0.49
	200	40.66	<sup>1</sup>	1.04	0.51
$p_j \in [1, 10]$	20	0.02	0.10	0.03	0.02
	40	0.21	0.41	0.05	0.12
	60	0.77	1.20	0.10	0.31
	100	3.65	4.23	0.23	0.63
	200	40.33	18.27	1.01	3.55

<sup>1</sup> To generate the TI models, in the case of  $p_j \in [1, 100]$ , it took 8 s for 20 jobs, 33 s for 40 jobs, 937 s for 60 jobs, and 11.185 s for 100 jobs. Therefore, the creation of the TI model for 200 jobs was ruled out.

Since the building time of a model is equivalent in all scheduling problems that are to be analyzed, the analysis of the TI formulation for the interval  $p_j \in [1, 100]$  is discarded. For any scheduling problem, preliminary tests for the interval  $p_j \in [1, 100]$  conclude that the TI models do not yield good feasible solutions for most of the cases.

As can be seen in Table 2, OPH and ODH are clearly the fastest formulations, equivalent to the resolution time of the LP relaxation. Although TI is slightly better than LO for  $p_j \in [1, 10]$ , the growth of the TI model may make its resolution unfeasible for the  $p_j \in [1, 100]$  distribution interval.

To evaluate the growth of the models, Table 3 shows how to calculate the size of the integer formulations for all four alternatives. In the TI formulation,  $T$  corresponds to the number of time periods. Table 4 displays the size for  $N = \{20, 40, 60, 100\}$  and shows that TI contains the smallest number of constraints for  $p_j \in [1, 10]$ , whereas the OPH formulation contains the smallest number of constraints for  $p_j \in [1, 100]$ . LO and ODH contain the smallest number of variables, but ODH considerably reduces the number of constraints.

**Table 3.** Formulation size expressions for  $1 || \Sigma w_j C_j$ .

Jobs		LO	SP	OPH	ODH	TI
$n$	Constraints	$n^3 - 5n^2/2 + 3n/2$	$n^2 + 3n$	$n^2/2 + 7n/2$	$n^2$	$T + n$
	Variables	$n^2$	$n^2 + 2n$	$2n^2$	$n^2$	$T(n - 1) + n$
	Integer variables	$n^2 - n$	$n^2$	$n^2$	$n^2 - n$	$T(n - 1) + n$

**Table 4.** Formulation size for  $1 || \Sigma w_j C_j$ .

Jobs		LO	SP	OPH	ODH	TI	
						$p_j \in [1, 10]$	$p_j \in [1, 100]$
20	Constraints	7030	460	270	400	136	922
	Variables	400	440	800	400	2320	18,580
	Integer variables	380	400	400	380	2320	18,580
40	Constraints	60,060	1720	940	1600	260	1874
	Variables	1600	1680	3200	1600	8800	73,360
	Integer variables	1560	1600	1600	1560	8800	73,360
60	Constraints	207,090	3780	2010	3600	391	2820
	Variables	3600	3720	7200	3600	19,860	165,580
	Integer variables	3540	3600	3600	3540	19,860	165,580
80	Constraints	975,150	10,300	5350	10,000	661	5087
	Variables	10,000	10,200	20,000	10,000	56,133	498,733
	Integer variables	9900	10,000	10,000	9900	56,133	498,733

5.3. Results for  $1 || L_{MAX}$

It was proved that the best formulation to solve the  $1 || L_{MAX}$  problem was the SP formulation [9], where the maximum execution time was one hour. In our study, the maximum time is set to  $n$  seconds. Table 5 shows the number of instances where the branch and bound execution finished (A), the number of instances where the optimal solution was reached (B), and the number of instances where each model obtained optimal or best feasible solutions compared to other formulations (C). Please note that we have considered the values of  $n = \{40, 60, 80, 100\}$ .

**Table 5.** Computational results for  $1 || L_{MAX}$  problem.

	$n$	LO			SP			TI			OPH			ODH		
		A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
$p_j \in [1, 100]$	40	21	21	21	36	36	36	-	-	-	22	36	36	23	36	36
	60	36	36	36	29	36	36	-	-	-	23	31	31	25	36	36
	80	24	24	24	27	34	34	-	-	-	12	19	19	21	36	36
	100	17	17	17	26	29	29	-	-	-	1	3	3	20	36	36
$p_j \in [1, 10]$	40	36	36	36	36	36	36	36	36	36	20	36	36	24	36	36
	60	35	35	35	36	36	36	36	36	36	20	28	28	21	36	36
	80	26	26	28	30	36	34	35	35	35	10	16	15	22	36	34
	100	20	20	20	27	30	30	25	25	25	1	6	6	22	34	34
Total		217	217	217	<b>247</b>	273	273	132	132	132	109	175	175	178	<b>286</b>	<b>286</b>

Greatest number of optimal solutions in each scenario (A, B and C) has been highlighted in bold.

Each row of Table 5 comprises 36 instances, i.e., 4 instances for each combination of [R, L]. Therefore, 288 problems have been solved. The ODH formulation achieves the best solutions, and  $p_j \in [1, 100]$  reaches optimal values for all instances. It can also be seen that the SP formulation is the one that most often completes the resolution, thus guaranteeing the optimal solution. Although the TI formulation shows very good behavior, it significantly decreases convergence with 100 jobs. LO never provides optimal solutions when it does not finish. Finally, OPH presents the worst performance here.

5.4. Results for  $1 \parallel \sum U_j$

Although ODH yields very good results for problem  $1 \parallel \sum U_j$ , the SP model provides the best overall performance (Table 6). However, if the results are analyzed based on parameter R (Table 7), ODH turns out to be the best model with R equal to 1 and 1.4, whereas its performance decreases with R equal to 0.6. At that value, SP works very well. In other words, SP is the best formulation for problems where delivery dates are very similar, but if we increase the range of delivery date values, ODH becomes the best formulation.

Table 6. Computational results for  $1 \parallel \sum U_j$  problem.

	n	LO			SP			TI			OPH			ODH		
		A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
$p_j \in [1, 100]$	40	12	24	24	31	36	36	-	-	-	17	28	28	23	36	36
	60	16	18	18	28	35	35	-	-	-	13	18	18	23	33	33
	80	11	11	11	26	36	36	-	-	-	5	7	7	21	31	31
	100	8	8	8	20	29	30	-	-	-	2	3	3	18	25	25
$p_j \in [1, 10]$	40	15	25	25	31	36	36	36	36	36	20	31	31	24	36	36
	60	12	13	13	30	36	36	28	29	29	11	15	15	20	34	34
	80	9	9	9	30	32	32	12	15	15	7	7	7	19	28	28
	100	8	8	8	24	32	33	7	8	8	3	3	3	20	28	29
Total		91	116	116	<b>220</b>	<b>272</b>	<b>274</b>	83	88	88	78	112	112	168	251	252

Greatest number of optimal solutions in each scenario (A, B and C) has been highlighted in bold.

Table 7. Computational results for  $1 \parallel \sum U_j$  problem with respect to parameter R.

R	LO			SP			TI			OPH			ODH		
	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
0.6	0	13	13	<b>89</b>	<b>94</b>	<b>96</b>	21	24	24	11	23	23	4	59	60
1	38	48	48	54	84	84	27	29	29	15	29	29	<b>68</b>	<b>96</b>	<b>96</b>
1.4	53	55	55	77	94	94	35	35	35	52	60	60	<b>96</b>	<b>96</b>	<b>96</b>
Total	91	116	116	220	272	274	83	88	88	78	112	112	168	251	252

Greatest number of optimal solutions in each scenario (A, B and C), as well as for each value of parameter R, has been highlighted in bold.

Table 6 also shows that TI loses convergence as the number of jobs increases, while the ODH formulation presents an overall better behavior for a greater interval of  $p_j$ .

5.5. Results for  $1 \parallel \sum T_j$

Henceforth, formulations are studied for the NP problems. Results are analyzed for  $n = \{40, 60, 80\}$  and are shown in Table 8. Since 4 instances for each combination [L, R] are considered, there is a total of 216 instances. The TI formulation presents the best overall results, provided that the formulation is feasible. If all ranges are considered, ODH presents

the best results by a considerable margin. Although the LO formulation comes in second place, it loses efficiency when the number of jobs increases, due to its constraints to avoid loops in the precedents of order  $O(n^3)$ .

**Table 8.** Computational results for  $1 || \Sigma T_j$  problem.

	<i>n</i>	LO			SP			TI			OPH			ODH		
		A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
$p_j \in [1, 100]$	40	10	27	27	10	25	25	-	-	-	5	26	26	22	33	33
	60	7	21	22	7	14	17	-	-	-	4	23	25	17	26	28
	80	6	17	17	5	7	15	-	-	-	0	15	17	9	23	24
$p_j \in [1, 10]$	40	7	27	27	7	36	36	36	36	36	5	24	24	21	31	31
	60	8	20	21	10	19	19	32	34	36	6	19	20	15	27	28
	80	6	19	19	6	15	15	30	36	36	0	10	10	11	27	27
Total		44	131	133	45	116	127	98	106	108	20	117	122	95	167	171

Greatest number of optimal solutions in each scenario (A, B and C) has been highlighted in bold.

As in the previous models, SP continues to have a direct formulation of the objective function with the use of positions. It is not necessary to calculate the delay of the job, but the delay of the position. However, SP formulation does not show good results, and there are no important differences regarding the length of the process times.

5.6. Results for  $1 || \Sigma w_j T_j$

The problem of minimizing the total weighted tardiness in single-machine scheduling is a well-known, strongly NP-hard problem [16]. Therefore, the resolution of instances until reaching an optimal solution becomes harder with size. We have decided to maintain the same instances as in the previous problem  $1 || \Sigma T_j$ .

In this problem, SP needs to calculate the delay of each job from the position delay, so this formulation will lose efficiency. Results are displayed in Table 9 and follow a similar trend as problem  $1 || \Sigma T_j$ .

**Table 9.** Computational results for  $1 || \Sigma w_j T_j$  problem.

	<i>n</i>	LO			SP			TI			OPH			ODH		
		A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
$p_j \in [1, 100]$	40	19	29	30	9	13	13	-	-	-	5	26	27	23	34	35
	60	9	21	23	7	10	13	-	-	-	5	16	17	18	30	29
	80	4	13	19	5	7	10	-	-	-	0	6	6	15	24	25
$p_j \in [1, 10]$	40	15	26	26	6	14	14	36	36	36	7	26	26	22	32	32
	60	9	21	21	10	12	12	35	36	36	7	15	15	18	26	26
	80	5	13	13	5	6	6	33	33	36	0	9	9	15	25	27
Total		61	123	132	42	62	68	104	105	108	24	98	100	111	171	174

Greatest number of optimal solutions in each scenario (A, B and C) has been highlighted in bold.

The TI formulation yields the best results. Regarding the rest of the formulations, ODH is the one presenting the best convergence. LO and OPH improve the results of the SP formulation, with LO being slightly better. There are no notable differences with respect to the length of the process times.

5.7. Results for  $P_m | \Sigma w_j C_j$

To minimize completion times with parallel machines, we have considered the number of jobs to be  $n = \{20, 40, 60\}$  and the number of machines  $m = \{2, 3, 5\}$ . Four instances for each combination of the processing time interval,  $\{p_j \in [1, 100], p_j \in [1, 10]\}$ ,  $n$ , and  $m$ , have been generated, leading to a total of 72 instances.

The TI formulation reaches the optimum in all 36 problems with  $p_j \in [1, 10]$ . If any range of  $p_j$  is considered, ODH outperforms the rest of the formulations, displaying the best performance in 45 problems. The OPH and LO formulations show similar results, with OPH being slightly better. Like the single-machine problem, SP exhibits very poor convergence. Since there were no differences regarding the length of the process times, results of both intervals have been integrated in Table 10. Therefore, each row contains eight problems, except for the TI formulation, which has only executed four problems.

Table 10. Computational results for  $P_m | \Sigma w_j C_j$  problem.

<i>m</i>	<i>n</i>	LO			SP			TI			OPH			ODH		
		A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
2	20	8	8	8	0	0	0	4	4	4	8	8	8	8	8	8
	40	0	0	0	0	0	0	4	4	4	0	0	0	0	0	4
	60	0	0	3	0	0	0	4	4	4	0	0	1	0	0	2
3	20	0	5	5	0	0	0	4	4	4	0	5	5	0	8	8
	40	0	0	0	0	0	0	4	4	4	0	0	4	0	0	4
	60	0	0	0	0	0	0	4	4	4	0	0	5	0	0	3
5	20	0	2	4	0	0	0	4	4	4	0	0	1	0	4	6
	40	0	0	2	0	0	0	4	4	4	0	0	2	0	0	4
	60	0	0	0	0	0	0	4	4	4	0	0	2	0	0	6
Total		8	15	22	0	0	0	<b>36</b>	<b>36</b>	36	8	13	28	8	20	<b>45</b>

Greatest number of optimal solutions in each scenario (A, B and C) has been highlighted in bold.

5.8. Results for  $P_m | \Sigma w_j T_j$

In the case of problem  $P_m | \Sigma w_j T_j$ , 648 problems are tested. As in the  $P_m | \Sigma w_j C_j$  problem, there were no differences regarding the interval of the process times, so the results of both intervals have been integrated in Table 11. Therefore, each row contains 72 problems, except for the TI formulation, which has only executed 36 problems in each row.

Like the previous problem, TI achieves optimal solutions for all problems. Although the results of the SP and ODH formulations are roughly similar, ODH provides a greater number of optimal solutions. Finally, OPH improves the LO formulation.

Table 11. Computational results for  $P_m | \Sigma w_j T_j$  problem.

<i>m</i>	<i>n</i>	LO			SP			TI			OPH			ODH		
		A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
2	20	38	45	47	8	22	22	36	36	36	25	38	40	46	58	60
	40	2	2	9	1	6	18	36	36	36	2	2	9	2	4	13
	60	2	2	2	11	11	22	36	36	36	4	4	5	0	0	20
3	20	15	21	25	6	13	19	36	36	36	10	19	22	23	32	45
	40	1	0	3	2	1	20	36	36	36	2	0	14	2	0	6
	60	0	0	10	9	9	25	36	36	36	3	2	7	0	0	3

Table 11. Cont.

m	n	LO			SP			TI			OPH			ODH		
		A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
5	20	14	23	27	6	13	18	36	36	36	9	20	25	20	34	41
	40	2	0	7	0	0	17	36	36	36	2	0	7	2	0	13
	60	0	0	0	6	3	17	32	36	36	1	1	3	2	1	24
Total		74	93	130	49	78	178	<b>320</b>	<b>324</b>	<b>324</b>	58	86	132	97	129	225

Greatest number of optimal solutions in each scenario (A, B and C) has been highlighted in bold.

5.9. Summary and Applications

Scheduling models are used in almost all production systems and environments. Their formulations are needed in many applications as a tool for the experimental study of heuristic allocation techniques and even as a way to achieve exact optimal solutions in environments with a restrained number of jobs.

As a summary of results, the total number of instances where each formulation obtained the optimal or the best feasible solution in each problem is shown in Figure 1. In other words, Figure 1 depicts the summation of the C column from Tables 5–11. It can be seen that ODH outperforms the other formulations in most of the problems. Regarding the  $P_m | \sum w_j T_j$  problem, the TI formulation achieves a great performance only in scenarios with a small number of jobs.

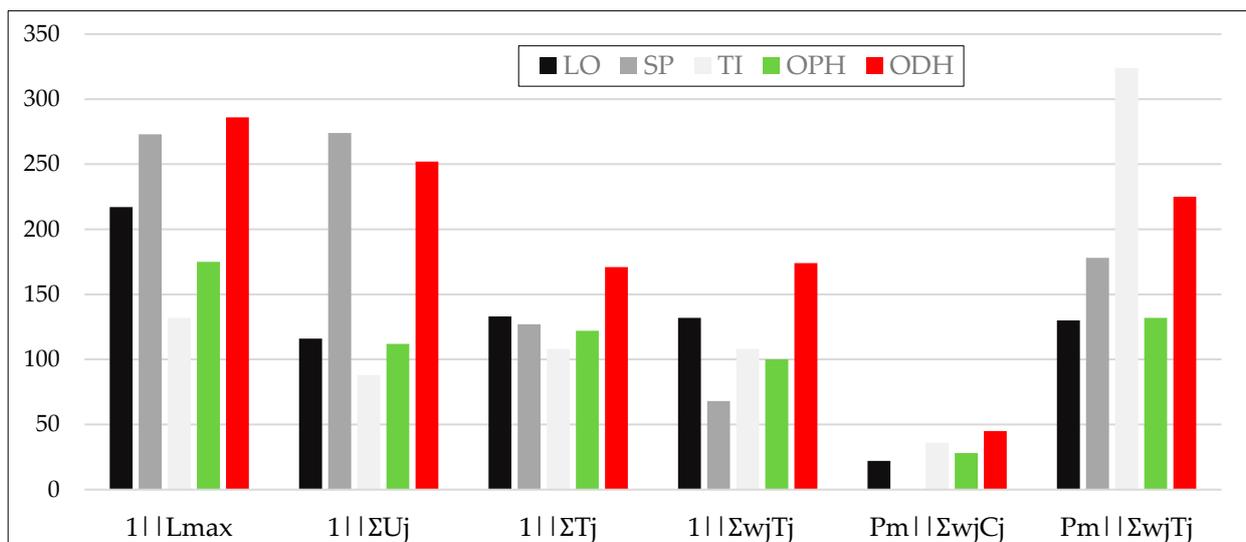


Figure 1. Total results for each formulation.

6. Conclusions

In this paper, two novel modeling approaches for scheduling problems have been presented by extending already existing modeling strategies. The strategy for the analysis has been to use short execution times: a number of seconds equal to the number of jobs times the number of machines. Afterward, we collected the number of completions reached and the number of optimal solutions, and instead of the average error, we have computed the number of times that each formulation achieved the best result amongst the five approaches.

The first formulation, OPH, based on the combination of the linear order and sequence position formulations, shows very good results in weighted completion time objectives. The second formulation, ODH, based on the combination of the linear order formulation and the

disjunctive formulation, yields the best results in our experimental study. Therefore, ODH should be considered the main formulation for further work and other study scenarios.

If only processing times belonging to the interval  $[1, 10]$  are assessed, the time index formulation produces the best results. However, when the processing time is extended to interval  $[1, 100]$ , the model drastically increases its size. For instance, it takes more than 15 min to generate a problem with 40 jobs and one machine. In addition, its results are very poor compared to the processing time.

Future lines of research involve extending the two novel formulations to flowshop environments. Concerning limitations of the research methodology, mathematical models can be inefficient in complex scenarios with a large number of jobs. However, as a tool for studying heuristic techniques, ODH achieves optimal solutions while showing a better behavior than previous formulations.

**Author Contributions:** Conceptualization, J.M.G.-S.; methodology, J.M.G.-S.; software, J.M.G.-S.; validation, J.M.G.-S. and P.M.; formal analysis, J.M.G.-S.; investigation, J.M.G.-S.; resources, J.M.G.-S.; writing—original draft preparation, J.M.G.-S.; writing—review and editing, P.M.; visualization, J.M.G.-S. and P.M.; supervision, J.M.G.-S. and P.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Dataset available on request from the authors.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

- Balas, E. On the facial structure of scheduling polyhedra. *Math. Program.* **1985**, *24*, 179–218. [[CrossRef](#)]
- Dyer, M.E.; Wolsey, L.A. Formulating the single machine sequencing problem with release dates as a mixed integer program. *Discret. Appl. Math.* **1990**, *26*, 255–270. [[CrossRef](#)]
- Nemhauser, G.L.; Savelsbergh, M.W.P. A cutting plane algorithm for the single machine scheduling problem with release times. In *Combinatorial Optimization: New Frontiers in Theory and Practice*; Akgul, M., Hamacher, H.W., Tufekci, S., Eds.; Springer: Berlin/Heidelberg, Germany, 1992; pp. 63–84. [[CrossRef](#)]
- Sousa, J.P.; Wolsey, L.A. A time-indexed formulation of non-preemptive single-machine scheduling problems. *Math. Program.* **1992**, *54*, 353–367. [[CrossRef](#)]
- Wagner, H.M. An integer programming model for machine scheduling. *Nav. Res. Logist.* **1959**, *6*, 131–140. [[CrossRef](#)]
- Lasserre, J.B.; Queyranne, M. Generic scheduling polyhedral and a new mixed-integer formulation for single-machine scheduling. In Proceedings of the 2nd IPCO (Integer Programming and Combinatorial Optimization) Conference, Pittsburgh, PA, USA, 25–27 May 1992; pp. 136–149.
- Keha, A.B.; Khowala, K.; Fowler, J.W. Mixed integer programming formulations for single machine scheduling problems. *Comput. Ind. Eng.* **2009**, *56*, 357–367. [[CrossRef](#)]
- Baker, K.R.; Keller, B. Solving the single-machine sequencing problem using integer programming. *Comput. Ind. Eng.* **2010**, *59*, 730–735. [[CrossRef](#)]
- Desrochers, M.; Laporte, G. Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. *Oper. Res. Lett.* **1991**, *10*, 27–36. [[CrossRef](#)]
- Chao, M.-T.; Lin, B.M.T. Scheduling of Software Test to Minimize the Total Completion Time. *Mathematics* **2023**, *11*, 4705. [[CrossRef](#)]
- Ouazene, Y.; Nguyen, N.-Q.; Yalaoui, F. Workload Balancing on Identical Parallel Machines: Theoretical and Computational Analysis. *Appl. Sci.* **2021**, *11*, 3677. [[CrossRef](#)]
- Kramer, A.; Dell’Amico, M.; Iori, M. Enhanced arc-flow formulations to minimize weighted completion time on identical parallel machines. *Eur. J. Oper. Res.* **2019**, *275*, 67–79. [[CrossRef](#)]
- Kramer, A.; Iori, M.; Lacomme, P. Mathematical formulations for scheduling jobs on identical parallel machines with family setup times and total weighted completion time minimization. *Eur. J. Oper. Res.* **2021**, *289*, 825–840. [[CrossRef](#)]
- Potts, C.N.; Van Wassenhove, L.N. An algorithm for single machine sequencing with deadlines to minimize total weighted completion time. *Eur. J. Oper. Res.* **1983**, *12*, 379–389. [[CrossRef](#)]

15. Hariri, A.M.A.; Potts, C.N. An algorithm for single machine sequencing with release dates to minimize total weighted completion time. *Discret. Appl. Math.* **1983**, *5*, 99–109. [[CrossRef](#)]
16. Lenstra, J.K.; Rinnooy Kan, A.H.G.; Brucker, P. Complexity of machine scheduling problems. *Ann. Discret. Math.* **1977**, *1*, 343–362. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.