



Article Developing New Bounds for the Performance Guarantee of the Jump Neighborhood for Scheduling Jobs on Uniformly Related Machines

Felipe T. Muñoz ^{1,*}, Guillermo Latorre-Núñez ¹, and Mario Ramos-Maldonado ²

- ¹ Departamento de Ingeniería Industrial, Facultad de Ingeniería, Universidad del Bío-Bío, Concepción 4051381, Chile; glatorre@ubiobio.cl
- ² Departamento de Ingeniería en Maderas, Facultad de Ingeniería, Universidad del Bío-Bío, Concepción 4051381, Chile; mramos@ubiobio.cl
- * Correspondence: fmunoz@ubiobio.cl

Abstract: This study investigates the worst-case performance guarantee of locally optimal solutions to minimize the total weighted completion time on uniformly related parallel machines. The investigated neighborhood structure is Jump, also called insertion or move. This research focused on establishing the local optimality condition expressed as an inequality and mapping that maps a schedule into an inner product space so that the norm of the mapping is closely related to the total weighted completion time of the schedule. We determine two new upper bounds for the performance guarantee, which take the form of an expression based on parameters that describe the family of instances: the speed of the fastest machine, the speed of the slowest machine, and the number of machines. These new bounds outperform the parametric upper bound previously established in the existing literature and enable a better understanding of the performance of the solutions obtained for the Jump neighborhood in this scheduling problem, according to parameters that describe the family of instances.

check for updates

Citation: Muñoz, F.T.; Latorre-Núñez, G.; Ramos-Maldonado, M. Developing New Bounds for the Performance Guarantee of the Jump Neighborhood for Scheduling Jobs on Uniformly Related Machines. *Mathematics* **2024**, *12*, 6. https:// doi.org/10.3390/math12010006

Academic Editors: Daniel A. Rossit and Frank Werner

Received: 20 October 2023 Revised: 13 November 2023 Accepted: 20 November 2023 Published: 19 December 2023



Copyright: © 2023 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/). **Keywords:** parallel machines; total weighted completion time; local search; jump neighborhood; performance guarantee

MSC: 90B35; 90C59; 68W40; 68M20

1. Introduction

Scheduling problems, which deal with the optimal allocation of limited resources to tasks to minimize or maximize certain objectives, have been a central research focus in the combinatorial optimization literature. The significance of these problems is evident, given their diverse range of practical applications, from job scheduling in manufacturing systems to resource management in complex projects. As Graham highlighted in their influential work [1], scheduling problems present intricate challenges, and their computational complexity makes them essential subjects of study for the optimization community.

Considering the NP-hard complexity of many of these problems, exact solutions frequently become impractical in real-world scenarios. Therefore, the development and analysis of approximate solution methods become imperative. In recent studies, scheduling problems in parallel machine environments have been investigated across various industries, including semiconductor manufacturing [2,3], metalworking [4], textile manufacturing [5], automotive manufacturing [6,7], chemical processes [8], offshore oil and gas extraction [9], as well as manufacturing scheduling for contingent new requirements [10–15]. Additionally, scheduling challenges within service-related sectors have been studied, encompassing healthcare [7,16,17], audit [18], transport and cross-docking [19,20], hospital waste collection [21], emergency response for forest fire extinguishing [22], and drone scheduling problems [23], among others.

1.1. Parallel Machine Environments

In problems related to the parallel machine scheduling environment, we are presented with a set of *n* jobs to be scheduled on a set of *m* machines. Typically, jobs are scheduled without interruption on a single machine, and each machine can process only one job at a time. Each job has a known weight (relative importance) and a processing requirement (processing time). Following the standard three-field scheduling notation [1,24], these problems are represented as $\alpha |\beta| \gamma$, where α describes the machine environment, β describes the job characteristics and constraints, and γ presents the objective function. The most straightforward scenario involves identical parallel machines (*P*), where the processing speeds, we enter the domain of the uniformly related parallel machine environment (*Q*). In contrast, when the processing time of a job depends arbitrarily on the machine where it will be processed, we have the environment of unrelated parallel machines (*R*).

A solution to the problems, or schedule, is an assignment and sequence of the jobs for each machine. Given a schedule, the completion time of job *j* can be determined, represented by C_j , with which two of the most studied objective functions can be established: the Weighted Total Completion Time, the objective of which is to minimize $\sum_j w_j C_j$, where $w_j \ge 0$ is the weight of the job *j*, and the Makespan, the objective of which is to determine a schedule that minimizes the maximum completion time of overall jobs, expressed as $C_{\text{max}} = \max_j \{C_j\}$.

In the context of computational complexity, achieving a solution to the parallel machine environment problem of minimizing the total completion time $(\sum C_j)$ can be accomplished in polynomial time. To be more specific, the problems $P||\sum C_j$ and $Q||\sum C_j$ can be efficiently solved using the Shortest Processing Time (SPT) rule, with complexities of $O(n \log n)$ and $O(n \log mn)$, respectively, [25,26]. Moreover, the $R||\sum C_j$ problem can be efficiently solved using appropriate bipartite matching techniques [27]. Nevertheless, if the goal is to minimize the total weighted completion time or the makespan, these problems present NPhard complexities [28]. This holds true even in scenarios with two identical machines [29,30]. Moreover, if the number of machines is included as part of the input, these problems become strongly NP-hard [26,31,32]. Given the inherent complexity of these problems, it is usual to look for approximate solution approaches, such as Polynomial Time Approximation Schemes (PTASs). For $P||\sum w_jC_j$ and $Q||\sum w_jC_j$ problems, PTASs are reported in [33,34]. However, for $R||\sum w_jC_j$ problems, the best-reported approximation is $3/2 - \delta$, where δ is a small non-negative constant [35,36]. Another commonly used approximate solution approach is Local Search.

1.2. Local Search

According to Williamson and Shmoys [37], the study of approximation algorithms provides a mathematically rigorous basis for the study of heuristics. Traditionally, heuristics and metaheuristics are empirically studied, demonstrating satisfactory performance. However, gaining a comprehensive understanding of their efficacy is essential. Thus, the study of performance guarantees bringing mathematical rigor to heuristics, enabling us to comprehend the quality of solutions obtained for all instances of the problem or to generate insights into the families of instances where the heuristic may not perform well.

Local search approaches are commonly employed for addressing scheduling problems, demonstrating notable empirical performance, but our understanding of their worst-case theoretical performance still needs to be improved. For a comprehensive review of performance guarantees and other theoretical considerations regarding local search across a broad spectrum of combinatorial problems, encompassing scheduling problems, readers are directed to [38,39]. The efficiency of a local algorithm depends on two critical factors: the size of the neighborhood and the quality of the local optima solution, as outlined by Ahuja [40]. One approach to assessing the quality of a local optimum is through worst-case analysis, which can be quantified using the performance guarantee.

$$pg(\mathcal{P}, \mathcal{N}) = \sup_{k \in \mathcal{I}} \sup_{\sigma \in \mathcal{L}_k} \left\{ \frac{cost(\sigma)}{opt(k)} \right\},\tag{1}$$

where \mathcal{P} represents the problem, \mathcal{N} represents the neighborhood structure, \mathcal{I} represents the set of all instances of \mathcal{P} , \mathcal{L}_k represents the set of all locally optimal solutions of instance k for neighborhood \mathcal{N} , and $cost(\sigma)$ and opt(k) are the values of the objective function for solution σ and optimal solution of instance k, respectively.

1.3. Aim and Scope

This study delves into analyzing the worst-case performance guarantee of solutions determined by local search approaches. Specifically, we investigate the Jump neighborhood, also known as move or insertion, where a job is relocated or reassigned from one machine to another until a predetermined stopping criterion is met. We focus on the scheduling problem of minimizing the total weighted completion time in a setting of uniformly related parallel machines, denoted as $Q||\sum w_j C_j$. We aim to establish an improved parametric upper bound for the performance guarantee.

1.4. Related Works

Regarding the performance guarantee of the studied problem, the literature sets a lower bound of 1.423 and an upper bound of 2.618. The lower bound is derived from an instance with three jobs and three machines [41]. In contrast, the upper bound is determined via the performance guarantee of problems $R||\sum w_jC_j$ and $Q|\mathcal{M}_j|\sum w_jC_j$, both problems being generalizations of the problem studied [42]. Additionally, the literature introduces a parametric upper bound, which establishes an upper bound for the performance guarantee based on parameters such as the speed of the fastest machine, the speed of the slowest machine, and the number of machines [41]. This expression enables the establishment of upper bounds for the performance guarantees lower than 2.618 for certain families of instances.

Table 1 displays the performance guarantee of the Jump neighborhood for scheduling jobs on parallel machine environments to minimize the total weighted completion time. The performance guarantee is tight for problems where a single value is reported. Specifically, this represents the worst-case performance guarantee value for a locally optimal solution within the Jump neighborhood. For problems where a range of values is reported, these values represent the lower and upper bounds for the worst-case performance guarantee. These open gaps highlight a research opportunity to establish a tight performance guarantee for the problem or narrow the gap between the upper and lower bounds. However, the challenge seems complicated since some gaps have remained open for over fifteen years, such as the gap of the $P||\sum w_iC_i$ problem [43].

The performance guarantee of locally optimal solutions for the Jump neighborhood has also been studied for makespan minimization. Specifically, the problem $P||C_{max}$ has been investigated in [44,45], the problem $Q||C_{max}$ in [45,46], and the problem $R||C_{max}$ in [45]. Problems involving machine eligibility restrictions were also studied for the P and Q environments in [47,48]. The analysis of performance guarantees in parallel machine environments has also extended to other neighborhood structures, primarily focusing on minimizing the makespan. These neighborhoods include lexjump [45,47–49], push [45,47], multi-exchange [45,50], and split [49]. Lexjump and push are polynomial-sized neighborhoods, while multi-exchange and split are exponential. Moreover, the efficiency of local search for the Jump neighborhood has also been studied [45,51].

| Problem | Jump Performance | References | |
|------------------------------------|--|------------|--|
| $R \sum w_iC_i$ | 2.618 | [42] | |
| $Q \mathcal{M}_i \sum w_iC_i^{-1}$ | 2.618 | [42] | |
| $P \mathcal{M}_i \sum w_i C_i$ | [1.75, 1.809] | [42] | |
| $Q \Sigma w_i C_i$ | [1.423, 2.618] | [41,42] | |
| $Q \sum w_j C_j^{2,3}$ | $ub=2\Big(1-rac{2}{m+2}\Big)\Big(rac{\mathrm{s}_m}{\mathrm{s}_1}+rac{1}{2m}\Big)$ | [41] | |
| $P \sum w_j C_j$ | $\begin{bmatrix} 6 \\ 5 \\ 2 \\ - \\ \frac{1}{2m} \end{bmatrix}$ | [43] | |
| $R \sum C_i$ | 2 | [42] | |
| $Q \mathcal{M}_i \sum C_i$ | 2 | [42] | |
| $P \mathcal{M}_j \sum C_j$ | [1.525, 1.618] | [42] | |

Table 1. Performance guarantee of the Jump neighborhood for scheduling jobs on parallel machine environments to minimize the total weighted completion time.

¹ \mathcal{M}_j denotes machine eligibility restrictions. ² s_m and s_1 represent the speed of the fastest and slowest machine, respectively, and *m* represents the number of machines. ³ *ub* represents the parametric upper bound for performance guarantee.

1.5. Our Results

Our main result establishes that the performance guarantee for locally optimal solutions under the Jump neighborhood for the $Q||\sum w_j C_j$ problem does not exceed the following upper bounds:

$$2 + \left(s_m - 1 - \frac{s_m}{m}\right) \mathbb{1}_{s_m > \frac{m}{m-1}},\tag{2}$$

$$\frac{1}{2}\left(\frac{s_m}{s_1} + s_m + 1 - \frac{s_m}{m}\right),\tag{3}$$

where s_m and s_1 represent the speed of the fastest and slowest machine, respectively, m represents the number of machines, and 1 represents the unit step function. These results improve upon the parametric upper bounds reported in [41]:

$$2\left(1-\frac{2}{m+2}\right)\left(\frac{s_m}{s_1}+\frac{1}{2m}\right).$$
(4)

Our proof technique is similar to the one employed in [41], as it establishes a local optimality condition for the solutions obtained using the Jump neighborhood. Subsequently, we utilize some properties of the solutions to the problem to establish an upper bound for the performance guarantee of the problem. The main difference in the development compared to [41] lies in the lower bound of the total weighted completion time for the optimal schedule. In [41], a lower bound is derived based on the inequality proposed by Eastman et al. [52]. In contrast, in this study, we define a lower bound based on a property obtained through a transformation. This transformation maps the set of feasible schedules of the problem to a specific inner product space, designed such that the norm closely corresponds with the total weighted completion time of the schedule.

The remainder of this paper is structured as follows. The scheduling problem and the local search neighborhood are introduced in Section 2. In Section 3, we present the properties of feasible and optimal solutions to the problem. The performance guarantee of the Jump neighborhood is studied in Section 4. Section 5 conducts the discussion. Finally, the main conclusions of this study are summarized in Section 6.

2. Preliminaries

This section offers a comprehensive description of the scheduling problem addressed in this study and the notation employed for its analysis. Furthermore, we present an overview of the studied neighborhood structure, which will aid in establishing the local optimality condition for the solution studies in this work.

2.1. Problem Statement

The problem addressed in this study consists of minimizing the total weighted completion time on uniform parallel machines. In scheduling notation [1,24], this problem is represented by $Q||\sum w_j C_j$. Let \mathcal{J} represent the set of n jobs and \mathcal{M} denote the set of $m \ge 2$ machines. For each job $j \in \mathcal{J}$, let p_j be the non-negative processing requirement, while w_j represents the non-negative weight of the job j. Each job must be scheduled without interruption on a single machine. Each machine can only process one job at a time. Let s_i denote the processing speed of machine $i \in \mathcal{M}$. Hence, if job j is assigned to machine i, a processing time p_j/s_i is required. Without loss of generality, we assume that the machines are indexed based on their speed, and we rescale the machine speeds so that:

$$s_1 \le s_2 \le \dots \le s_m,\tag{5}$$

$$\sum_{i \in \mathcal{M}} s_i = m. \tag{6}$$

Following the notation in [41,42], a schedule, denoted as v, represents a solution to the problem and establishes the assignment of jobs to machines. Let v_j indicate the machine to which job j is assigned. More precisely, $v_j = i$ indicates that job j is assigned to machine i in schedule v.

The sequence in which jobs assigned to machine *i* should be processed is determined by the Weighted Shortest Processing Time (WSPT) rule [53] (Thm. 3). This rule arranges jobs in decreasing order of the w_j/p_j ratio. In the case of ties in the ratio, they are broken arbitrarily. For clarity and to avoid confusion, we use the \prec , \succ , \preceq , and \succeq notation for describing the precedence relationship between jobs and the job itself induced via the WSPT rule. Then, representing the set of jobs assigned to machine *i* in schedule *v* as $\mathcal{J}_i(v)$, for $v_j = i$, we can express the value of the completion time of job *j* as:

$$C_{j}(\boldsymbol{v}) = \sum_{\substack{k \in \mathcal{J}_{i}(\boldsymbol{v}) \\ k \leq j}} \frac{p_{k}}{s_{i}} = \sum_{\substack{k \in \mathcal{J}_{v_{j}}(\boldsymbol{v}) \\ k \neq j}} \frac{p_{k}}{s_{v_{j}}}.$$
(7)

The total weighted completion time and the weighted sum of processing times for schedule v are defined as follows:

$$C(\boldsymbol{v}) = \sum_{j \in \mathcal{J}} w_j C_j(\boldsymbol{v}) = \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{J}_i(\boldsymbol{v})} w_j C_j(\boldsymbol{v}),$$
(8)

$$\eta(\boldsymbol{v}) = \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{J}_i(\boldsymbol{v})} \frac{w_j p_j}{s_i} = \sum_{j \in \mathcal{J}} \frac{w_j p_j}{s_{v_j}}.$$
(9)

With the previous definitions, we have the following identities:

$$C(\boldsymbol{v}) = \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{J}_i(\boldsymbol{v})} \sum_{\substack{k \in \mathcal{J}_i(\boldsymbol{v}) \\ k \preceq j}} \frac{w_j p_k}{s_i} = \sum_{j \in \mathcal{J}} \sum_{\substack{k \in \mathcal{J}_{v_j}(\boldsymbol{v}) \\ k \prec i}} \frac{w_j p_k}{s_{v_j}} = \eta(\boldsymbol{v}) + \sum_{j \in \mathcal{J}} \sum_{\substack{k \in \mathcal{J}_{v_j}(\boldsymbol{v}) \\ k \prec i}} \frac{w_j p_k}{s_{v_j}}, \quad (10)$$

$$C(\boldsymbol{v}) = \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{J}_i(\boldsymbol{v})} \sum_{\substack{k \in \mathcal{J}_i(\boldsymbol{v}) \\ k \succeq j}} \frac{w_k p_j}{s_i} = \sum_{j \in \mathcal{J}} \sum_{\substack{k \in \mathcal{J}_{v_j}(\boldsymbol{v}) \\ k \succeq j}} \frac{w_k p_j}{s_{v_j}} = \eta(\boldsymbol{v}) + \sum_{j \in \mathcal{J}} \sum_{\substack{k \in \mathcal{J}_{v_j}(\boldsymbol{v}) \\ k \succ j}} \frac{w_k p_j}{s_{v_j}}.$$
 (11)

2.2. Jump Neighborhood

The Jump neighborhood, also called insertion or move, is a polynomial-size neighborhood [49]. A Jump move is characterized by relocating a single job from one machine to another. The success of a Jump move is determined by the reduction in the objective function (total weighted completion time). Given a solution, if it is impossible to make Jump

moves that improve the value of the objective function, this solution is a local optimum. We call this solution Jump-Opt.

Figure 1 illustrates a Jump move of a schedule denoted as x. In this scheme, job j, currently assigned to machine i, is moved or reassigned to machine h. The figure also indicates the sets of jobs scheduled before and after job j on both machines. Consider $\delta_j(x)$ as the reduction in the total weighted completion time when job j is excluded from machine $x_j = i$. Furthermore, let $\delta'_j(x, h)$ denote the increase in the total weighted completion time if job j is reassigned to machine h. Thus,

$$\delta_j(\mathbf{x}) = w_j C_j(\mathbf{x}) + \sum_{\substack{k \in \mathcal{J}_{x_j}(\mathbf{x}) \\ k \succ j}} \frac{w_k p_j}{s_{x_j}},$$
(12)

$$\delta'_{j}(\mathbf{x},h) = \frac{w_{j}p_{j}}{s_{h}} + \sum_{\substack{k \in \mathcal{J}_{h}(\mathbf{x}) \\ k \prec j}} \frac{w_{j}p_{k}}{s_{h}} + \sum_{\substack{k \in \mathcal{J}_{h}(\mathbf{x}) \\ k \succ j}} \frac{w_{k}p_{j}}{s_{h}}.$$
 (13)

Consequently, the schedule *x* will be a Jump-Opt solution if, and only if,

$$\delta_j(\mathbf{x}) \le \delta'_j(\mathbf{x}, h) \text{ for all } j \in \mathcal{J}, \ h \in \mathcal{M}.$$
 (14)

| Schedule of machine <i>i</i> : | $k \in \mathcal{J}_i(\boldsymbol{x}) : k \prec j$ | j | $k \in \mathcal{J}_i(\boldsymbol{x}) : k \succ j$ | | |
|--------------------------------|---|---------------------|---|--|--|
| | Jump move | | | | |
| Schedule of machine <i>h</i> : | $k \in \mathcal{J}_h(\boldsymbol{x}) : k \prec j$ | $k \in \mathcal{C}$ | $\mathcal{T}_h(\boldsymbol{x}): k \succ j$ | | |

Figure 1. Schematic for a Jump move.

3. Properties of the Optimal Schedules

This section describes some properties of feasible and optimal solutions to the $Q||\sum w_j C_j$ problem. Guided by the insights of Cole et al. [54], we develop mapping from the set of schedules to a specific inner product space. This mapping is designed such that the norm closely corresponds to the total weighted completion time of the schedule. Let $\varphi : \mathcal{M}^{\mathcal{J}} \to L_2([0,\infty))^{\mathcal{M}}$ be the function that associates every feasible schedule v with a vector of functions, with one for each machine. If $f = \varphi(v)$, then for each machine $i \in \mathcal{M}$, we define:

$$f_i(y) = \sum_{j \in \mathcal{J}_i(v): \ \frac{p_j}{w_i} \ge y} \frac{w_j}{\sqrt{s_i}}.$$
(15)

Lemma 1. Let v be a schedule of the $Q||\sum w_i C_i$ problem, and let $f = \varphi(v)$. Then,

$$\|\varphi(v)\|^{2} = 2C(v) - \eta(v).$$
(16)

Proof. For $f = \varphi(v)$, the norm is calculated as:

$$\begin{split} \|\varphi(\boldsymbol{v})\|^2 &= \sum_{i \in \mathcal{M}} \int_0^\infty f_i(y)^2 dy = \sum_{i \in \mathcal{M}} \int_0^\infty \sum_{\substack{j \in \mathcal{J}_i(\boldsymbol{v}): \ \frac{p_j}{w_j} \ge y}} \frac{w_j}{\sqrt{s_i}} \sum_{\substack{k \in \mathcal{J}_i(\boldsymbol{v}): \ \frac{p_k}{w_k} \ge y}} \frac{w_k}{\sqrt{s_i}} \, dy \\ &= \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{J}_i(\boldsymbol{v})} \sum_{\substack{k \in \mathcal{J}_i(\boldsymbol{v})}} \frac{w_j w_k}{s_i} \int_0^\infty \mathbb{1}_{y \le \frac{p_j}{w_j}} \mathbb{1}_{y \le \frac{p_k}{w_k}} \, dy \\ &= \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{J}_i(\boldsymbol{v})} \sum_{\substack{k \in \mathcal{J}_i(\boldsymbol{v})}} \frac{w_j p_k}{s_i} \mathbb{1}_{\frac{p_k}{w_k} \le \frac{p_j}{w_j}} + \frac{w_k p_j}{s_i} \mathbb{1}_{\frac{p_k}{w_k} \ge \frac{p_j}{w_j}}. \end{split}$$

By utilizing the \prec , \succ , \preceq , and \succeq notation, as introduced through the application of the WSPT rule, we obtain

$$\|\varphi(v)\|^2 = \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{J}_i(v)} \sum_{\substack{k \in \mathcal{J}_i(v) \\ k \prec j}} \frac{w_j p_k}{s_i} + \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{J}_i(v)} \sum_{\substack{k \in \mathcal{J}_i(v) \\ k \succ j}} \frac{w_k p_j}{s_i}.$$

By using Equations (9)–(11), the proof is concluded. \Box

For subsequent analysis, it is necessary to quantify the total weighted completion time of a schedule in which all jobs are assigned to a single machine that operates at a speed equal to 1. The total weighted completion time of this schedule is

$$Z_1 = \sum_{j \in \mathcal{J}} \sum_{\substack{k \in \mathcal{J} \\ k \leq j}} w_j p_k = \sum_{j \in \mathcal{J}} \sum_{\substack{k \in \mathcal{J} \\ k \geq j}} w_k p_j.$$
(17)

Another solution of particular interest is the one where all jobs are assigned to a single machine that operates at speed m. Remember that according to assumption (6), the sum of the speeds of the machines is equal to m. We represent this schedule by z, and its total weighted completion time is

$$C(z) = \frac{Z_1}{m}.$$
(18)

In the following lemma, φ mapping is utilized to establish another property of solutions to the problem $Q||\sum w_i C_i$.

Lemma 2. Let v be a schedule for the $Q||\sum w_j C_j$ problem, where $\sum_{i \in \mathcal{M}} s_i = m$, and let z denote a schedule in which all jobs are assigned to a machine operating at speed m (the total number of machines). For $f = \varphi(v)$ and $f^z = \varphi(z)$, it holds that

$$\|\varphi(z)\|^2 \le \|\varphi(v)\|^2.$$
 (19)

Before proceeding with the proof of Lemma 2, a graphical concept of its demonstration is presented. Figure 2 depicts the graph associated with the mapping φ for an instance with n = 4 jobs and m = 2 machines. For this instance, we have the schedule v = (1, 2, 1, 2), which indicates that jobs j_1 and j_3 are assigned to machine 1, while jobs j_2 and j_4 are assigned to machine 2. The representation of this schedule is presented in Figure 2b,c. Here, it can be observed that each job is represented by a rectangle, with its height being determined by $w_j/\sqrt{s_i}$, and its length being determined by p_j/w_j . The order in which the jobs are arranged is defined by the WSPT rule. Specifically, they are ordered in decreasing order concerning the ratio w_j/p_j . In Figure 2b,c, the value of the function related to the mapping φ for each machine is depicted with red lines, denoted as $f_1(y)$ and $f_2(y)$, for each machine, respectively. In Figure 2a, the sum value of these functions is depicted with a dotted red line. The value of the function related to the mapping φ for the schedule where all jobs are assigned to a single machine operating at speed *m* is depicted by a blue line. It is evident from this graph that the value of the function for this schedule is always less than or equal to $f_1(y) + f_2(y)$ for any value of $y \ge 0$. This result can be generalized for any instance. Next, we proceed with the proof of the lemma.



Figure 2. A mapping of schedules *z* and *v* for an instance involving four jobs and two machines.

Proof of Lemma 2. By applying the mapping φ to schedule *z*, we have a single vector of functions, with

$$f^{z}(y) = \sum_{j \in \mathcal{J}: \ \frac{p_{j}}{w_{j}} \ge y} \frac{w_{j}}{\sqrt{m}}$$

The norm for f^z is,

$$\begin{split} \|\varphi(\mathbf{z})\|^2 &= \int_0^\infty f^z(y)^2 dy = \int_0^\infty \left(\sum_{\substack{j \in \mathcal{J}: \ \frac{p_j}{w_j} \ge y}} \frac{w_j}{\sqrt{m}}\right)^2 dy \\ &= \int_0^\infty \frac{1}{m} \left(\sum_{\substack{j \in \mathcal{J}: \ \frac{p_j}{w_j} \ge y}} w_j\right)^2 dy = \int_0^\infty \frac{1}{m} \left(\sum_{\substack{i \in \mathcal{M} \ j \in \mathcal{J}_i(\mathbf{v}): \ \frac{p_j}{w_j} \ge y}} w_j\right)^2 dy. \end{split}$$

In the last equality, based on the schedule v, the set of all jobs is represented by m disjoint sets of jobs. Thus, for this final expression, the Cauchy–Bunyakovsky–Schwarz inequality is utilized to conclude that:

$$\|\varphi(z)\|^2 \le \int_0^\infty \sum_{i \in \mathcal{M}} \left(\sum_{j \in \mathcal{J}_i(v): \ \frac{p_j}{w_j} \ge y} w_j \right)^2 dy = \sum_{i \in \mathcal{M}} \int_0^\infty \left(\sum_{j \in \mathcal{J}_i(v): \ \frac{p_j}{w_j} \ge y} w_j \right)^2 dy.$$
(20)

However, the norm for f is:

$$\|\varphi(\boldsymbol{v})\|^{2} = \sum_{i \in \mathcal{M}} \int_{0}^{\infty} f(y)^{2} dy = \sum_{i \in \mathcal{M}} \int_{0}^{\infty} \left(\sum_{\substack{j \in \mathcal{J}_{i}(\boldsymbol{v}): \ w_{j} \geq y}} \frac{w_{j}}{\sqrt{s_{i}}} \right)^{2} dy$$
$$= \sum_{i \in \mathcal{M}} \frac{1}{s_{i}} \int_{0}^{\infty} \left(\sum_{\substack{j \in \mathcal{J}_{i}(\boldsymbol{v}): \ w_{j} \geq y}} w_{j} \right)^{2} dy.$$
(21)

To simplify the notation, we define:

$$\alpha_i = \int_0^\infty \left(\sum_{j \in \mathcal{J}_i(\boldsymbol{v}): \ \frac{p_j}{w_j} \ge y} w_j \right)^2 dy.$$
(22)

Without loss of generality, we assume that the job processing requirements and weights take values such that:

$$\alpha_i \leq 1, \ \forall i \in \mathcal{M}.$$
 (23)

Next, we define the difference between $\|\varphi(z)\|^2$ and $\|\varphi(v)\|^2$ and utilize Equations (20)–(23):

$$L = \|\varphi(\mathbf{z})\|^2 - \|\varphi(\mathbf{v})\|^2 \le \sum_{i \in \mathcal{M}} \alpha_i \left(1 - \frac{1}{s_i}\right) \le \sum_{i \in \mathcal{M}} \left(1 - \frac{1}{s_i}\right).$$
(24)

To conclude the proof, it is necessary to demonstrate that the right-hand side of Equation (24) is less than or equal to zero. To achieve this, we formulate the following problem:

$$\max_{s \in \mathbb{R}_{+}^{m}} \left\{ \sum_{i \in \mathcal{M}} \left(1 - \frac{1}{s_{i}} \right), \text{ subject to } \sum_{i \in \mathcal{M}} s_{i} = m \right\}.$$
(25)

Note that the objective function of Problem (25) is concave, and the constraint is linear. Hence, applying the Lagrange method enables the identification of a global optimum for the problem. The Lagrangian function for Problem (25) is:

$$\mathcal{L}(\boldsymbol{s},\lambda) = \sum_{i \in \mathcal{M}} \left(1 - \frac{1}{s_i}\right) - \lambda\left(\sum_{i \in \mathcal{M}} s_i - m\right).$$

By solving the first-order necessary condition, we find the optimal solution: $\lambda = 1$ and $s_i = 1$, $\forall i \in \mathcal{M}$. This implies that the objective function value in Problem (25) equals zero, thereby demonstrating that

$$L = \|\varphi(z)\|^2 - \|\varphi(v)\|^2 \le 0.$$

The following theorem is established by applying Lemma 2 to the optimal schedule.

Theorem 1. For an optimal schedule x^* of the $Q||\sum w_j C_j$ problem, where $\sum_{i \in \mathcal{M}} s_i = m$, the following expression holds:

$$\frac{2Z_1}{m} - \sum_{j \in \mathcal{J}} \frac{w_j p_j}{m} \le 2C(\boldsymbol{x}^*) - \eta(\boldsymbol{x}^*).$$
(26)

Proof. The proof comes from Lemmas 1 and 2 and Equation (18). \Box

Next, we introduce an additional property that will be used to establish the main result of this study.

Lemma 3. For any schedule v of the $Q||\sum w_j C_j$ problem, the weighted sum of processing times satisfies

$$\sum_{j \in \mathcal{J}} \frac{w_j p_j}{s_m} \le \eta(\boldsymbol{v}) \le \sum_{j \in \mathcal{J}} \frac{w_j p_j}{s_1}.$$
(27)

Proof. The proof is established from Equation (9) and assumption (5). \Box

4. Performance Guarantee

The following two lemmas provide parametric upper bounds for the performance guarantee of locally optimal solutions for the Jump neighborhood. Here, x^* represents the optimal schedule, and x represents the Jump-Opt schedule.

Lemma 4. Given an instance of the $Q||\sum w_j C_j$ problem, where $\sum_{i \in M} s_i = m$, the performance guarantee of Jump-Opt solutions is, at most,

$$2 + \left(s_m - 1 - \frac{s_m}{m}\right) \mathbb{1}_{s_m > \frac{m}{m-1}}.$$
(28)

Proof. From the local optimality condition, Equation (14), we have:

$$w_j C_j(\boldsymbol{x}) + \sum_{\substack{k \in \mathcal{J}_{x_j}(\boldsymbol{x}) \\ k \succ j}} \frac{w_k p_j}{s_{x_j}} \leq \frac{w_j p_j}{s_h} + \sum_{\substack{k \in \mathcal{J}_h(\boldsymbol{x}) \\ k \prec j}} \frac{w_j p_k}{s_h} + \sum_{\substack{k \in \mathcal{J}_h(\boldsymbol{x}) \\ k \succ j}} \frac{w_k p_j}{s_h}, \forall j \in \mathcal{J}, h \in \mathcal{M}.$$

Next, we multiply both sides of the inequality by s_h/m and sum over all $h \in \mathcal{M}$. Note that, as established in Equation (6), $\sum_h s_h/m = 1$. Consequently, we derive the following valid inequality:

$$\begin{split} w_j C_j(\mathbf{x}) + \sum_{\substack{k \in \mathcal{J}_{x_j}(\mathbf{x}) \\ k \succ j}} \frac{w_k p_j}{s_{x_j}} &\leq w_j p_j + \sum_{\substack{h \in \mathcal{M} \\ k \prec j}} \sum_{\substack{k \in \mathcal{J}_h(\mathbf{x}) \\ k \prec j}} \frac{w_j p_k}{m} + \sum_{\substack{h \in \mathcal{M} \\ k \prec j}} \sum_{\substack{k \in \mathcal{J} \\ k \succ j}} \frac{w_k p_j}{m} \\ &= w_j p_j + \sum_{\substack{k \in \mathcal{J} \\ k \preceq j}} \frac{w_j p_k}{m} + \sum_{\substack{k \in \mathcal{J} \\ k \succeq j}} \frac{w_k p_j}{m} - \frac{2w_j p_j}{m}, \, \forall j \in \mathcal{J}. \end{split}$$

By summing over all $j \in \mathcal{J}$ and utilizing Equations (11) and (17), while grouping certain terms, we obtain the following expression:

$$2C(\mathbf{x}) \leq \eta(\mathbf{x}) + \left(1 - \frac{2}{m}\right) \sum_{j \in \mathcal{J}} w_j p_j + \frac{2Z_1}{m}.$$

Utilizing Theorem 1, it follows that:

$$2C(\mathbf{x}) \le \eta(\mathbf{x}) + \left(1 - \frac{1}{m}\right) \sum_{j \in \mathcal{J}} w_j p_j + 2C(\mathbf{x}^*) - \eta(\mathbf{x}^*).$$
⁽²⁹⁾

Using Equation (10) and Lemma 3, we have $\eta(\mathbf{x}) \leq C(\mathbf{x})$, and $\sum_{j \in \mathcal{J}} w_j p_j \leq s_m \eta(\mathbf{x}^*)$, respectively. Therefore,

$$C(\boldsymbol{x}) \leq 2C(\boldsymbol{x}^*) + \eta(\boldsymbol{x}^*) \Big(s_m - \frac{s_m}{m} - 1 \Big).$$
(30)

From Equation (30), it can be determined that there are two cases depending on the sign of the term that multiplies $\eta(x^*)$. The term is negative for $s_m \leq m/(m-1)$. In this case, we have

$$C(\mathbf{x}) \leq 2C(\mathbf{x}).$$

Conversely, if the term that multiplies $\eta(x^*)$ is positive, and given that $\eta(x^*) \leq C(x^*)$, we have

$$C(\mathbf{x}) \leq C(\mathbf{x}^*) \left(1 + s_m - \frac{s_m}{m}\right).$$

Finally, the proof is concluded by isolating $C(x)/C(x^*)$. \Box

Lemma 5. Given an instance of the $Q||\sum w_j C_j$ problem, where $\sum_{i \in M} s_i = m$, the performance guarantee of Jump-Opt solutions is, at most,

$$\frac{1}{2}\left(\frac{s_m}{s_1} + s_m + 1 - \frac{s_m}{m}\right).\tag{31}$$

Proof. This proof begins with Equation (29):

$$2C(\mathbf{x}) \leq \eta(\mathbf{x}) + \left(1 - \frac{1}{m}\right) \sum_{j \in \mathcal{J}} w_j p_j + 2C(\mathbf{x}^*) - \eta(\mathbf{x}^*).$$

According to Lemma 3, we deduce that $\eta(x) \leq \sum_{j \in \mathcal{J}} \frac{w_j p_j}{s_1}$. Therefore,

$$2C(\mathbf{x}) \leq \left(1 - \frac{1}{m} + \frac{1}{s_1}\right) \sum_{j \in \mathcal{J}} w_j p_j + 2C(\mathbf{x}^*) - \eta(\mathbf{x}^*).$$

$$2C(\mathbf{x}) \leq \left(s_m - rac{s_m}{m} + rac{s_m}{s_1} - 1
ight)\eta(\mathbf{x}^*) + 2C(\mathbf{x}^*).$$

Equation (10) makes it evident that $\eta(x^*) \leq C(x^*)$. Further, considering the non-negativity of the term multiplying $\eta(x^*)$, we have:

$$2C(\mathbf{x}) \leq \left(s_m - \frac{s_m}{m} + \frac{s_m}{s_1} + 1\right)C(\mathbf{x}^*).$$

Finally, the proof is concluded by isolating $C(x)/C(x^*)$. \Box

The subsequent lemma illustrates that the parametric upper bound introduced in Lemma 5 provides a tighter upper bound compared to the one presented in [41].

Lemma 6. The proposed parametric upper bound for the performance guarantee of Jump-Opt solutions for the $Q||\sum w_iC_i$ problem is better than Muñoz and Pinochet parametric upper bound.

Proof. By considering Equation (4) and Theorem 5, we determine the difference between the performance guarantees. Let *D* represent this difference:

$$D = 2\left(1 - \frac{2}{m+2}\right)\left(\frac{s_m}{s_1} + \frac{1}{2m}\right) - \frac{1}{2}\left(\frac{s_m}{s_1} + s_m + 1 - \frac{s_m}{m}\right)$$
$$= \frac{s_m}{s_1}\left(\frac{3m-2}{2(m+2)} - \frac{m}{2(m+2)}\frac{s_1}{s_m} - \frac{(m-1)}{2m}s_1\right).$$

To establish the lemma, it is sufficient to demonstrate that $D \ge 0$. Given that s_1/s_m and s_1 are upper-bounded by 1, we establish the following inequality:

$$D \ge \frac{s_m}{s_1} \left(\frac{3m-2}{2(m+2)} - \frac{m}{2(m+2)} - \frac{(m-1)}{2m} \right) = \frac{s_m}{s_1} \left(\frac{m^2 - 3m + 2}{2m(m+2)} \right) = \frac{s_m}{s_1} \frac{(m-1)(m-2)}{2m(m+2)}$$

This final expression shows that $D \ge 0$ holds true for all $m \ge 2$. \Box

In the following theorem, we present our main result, utilizing the parametric upper bounds established in Lemmas 4 and 5, in conjunction with the constant performance guarantee for the $Q|\mathcal{M}_j| \sum w_j C_j$ problem (refer to Table 1). It is important to note that the $Q|\mathcal{M}_j| \sum w_j C_j$ problem is a generalization of the $Q|| \sum w_j C_j$ problem. In the $Q|\mathcal{M}_j| \sum w_j C_j$ problem, the presence of machine eligibility restrictions is indicated by \mathcal{M}_j , meaning each job *j* can be processed by a subset $\mathcal{M}_j \subseteq \mathcal{M}$ of machines. This problem is considered a generalization since $|\mathcal{M}_j|$ can be equal to *m* for all jobs. Consequently, the performance guarantee for Jump-Opt solutions in the $Q|\mathcal{M}_j| \sum w_j C_j$ problem serves as an upper bound for the performance guarantee of Jump-Opt solutions in the $Q||\sum w_j C_j$ problem.

Theorem 2. Given an instance of the $Q||\sum w_jC_j$ problem, where $\sum_{i\in\mathcal{M}} s_i = m$, the performance guarantee of Jump-Opt solutions is, at most,

$$\min\left\{2.618, \ 2 + \left(s_m - 1 - \frac{s_m}{m}\right)\mathbb{1}_{s_m > \frac{m}{m-1}}, \ \frac{1}{2}\left(\frac{s_m}{s_1} + s_m + 1 - \frac{s_m}{m}\right)\right\}.$$
(32)

5. Discussion

Based on the results presented in Section 4, it becomes evident that the bound introduced in Muñoz and Pinochet [41] is outperformed by the bound proposed in Lemma 5. This performance difference demonstrated in Lemma 6, is the key reason why the upper bound from [41] is not incorporated into Theorem 2.

13 of 16

Next, we will discuss the complementarity of the three upper bounds presented in Theorem 2. We will examine this complementarity in the context of two machines and the scenario where *m* tends to infinity. We will use ub_2 and ub_3 to refer to the upper bounds of Lemmas 4 and 5, respectively.

For environments with m = 2 machines and given that $s_1 + s_2 = 2$, we have:

$$ub_{2} = 2 + \left(s_{2} - 1 - \frac{s_{2}}{2}\right)\mathbb{1}_{s_{2}>2} = 2,$$

$$ub_{3} = \frac{1}{2}\left(\frac{s_{2}}{s_{1}} + \frac{s_{2}}{2} + 1\right) = \frac{1}{s_{1}} + \frac{1}{2} - \frac{s_{1}}{4}.$$

The first observation for this case is that the fixed upper bound of 2.618 is dominated by the constant value of 2 provided by ub_2 . To assess the complementarity of ub_2 and ub_3 , we introduce the difference $R_2 = ub_3 - ub_2$. A positive value of R_2 implies that the bound ub_2 is tighter than ub_3 , while a negative value implies the opposite. Thus,

$$R_2 = \frac{1}{s_1} - \frac{s_1}{4} - \frac{3}{2} = \frac{(\sqrt{13} - 3 - s_1)(s_1 + 3 + \sqrt{13})}{4s_1}$$

Note that $R_2 \ge 0$ if, and only if, $s_1 \le \sqrt{13} - 3 \approx 0.6056$. Then, ub_2 is a better bound than ub_3 for $s_1 \le 0.6056$, while for $s_1 > 0.6056$, ub_3 is a better bound than ub_2 .

For $m \to \infty$, it should be noted that the condition of ub_2 , $s_m > m/(m-1) \to 1$, holds true consistently in a uniform parallel machines environment. Thus,

$$ub_2 = s_m + 1,$$

 $ub_3 = \frac{1}{2} \left(\frac{s_m}{s_1} + s_m + 1 \right).$

Analogously to the analysis of the previous case, we use the difference to analyze the complementarity of the upper bounds. Let $R_m = ub_3 - ub_2$ be the difference for this case,

$$R_m = \frac{1}{2} \left(\frac{s_m}{s_1} - s_m - 1 \right).$$

The sign that R_m takes depends on the values of s_1 and s_m . Therefore, both of the two upper bounds must be considered. In other words, none of the upper bounds is dominated by the other.

To illustrate the complementarity of the three upper bounds included in Theorem 2, we provide examples. Table 2 presents five intentional parameter combinations for the $Q||\sum w_jC_j$ problem, along with the values obtained from the upper bounds ub_2 and ub_3 , in addition to the upper bound for the generalization of the problem [42], denoted as ub_1 . The results presented provide evidence of the usefulness of all the upper bounds, contingent upon the specific parameter combinations. These examples underscore how the Theorem 2 enables the establishment of better upper bounds than those attainable through each upper bound individually. Note that in combinations 1 to 3, $s_5 > m/(m-1) = 1.25$. In contrast, in combinations 4 and 5, $s_5 \leq 1.25$. This discrepancy has implications for determining ub_2 .

Finally, we extend the result of Lemma 5 to a particular case. The problem under study, $Q||\sum w_jC_j$, is a generalization of the problem in an environment with identical parallel machines, represented by $P||\sum w_jC_j$. For this problem, Brueggemann et al. [43] determined that the performance guarantee for solutions obtained using the Jump neighborhood lies within the interval [1.2, (3m - 1)/2m]. Regarding this result, it can be observed that the parametric upper bound for the performance guarantee for the Jump neighborhood proposed in Lemma 5 coincides with the value reported in [43] when $s_1 = s_m = 1$. Note that these values refer to the scenario of identical machines operating at unitary speed.

| | Machine Speeds | s_{5}/s_{1} | Bounds | Upper Bound | Best Bound |
|---------|---|---------------|--|-------------|-----------------|
| Comb. 1 | $s_1 = 0.050$ $s_2 = 0.100$ $s_3 = 0.100$ $s_4 = 0.150$ $s_5 = 4.600$ | 92 | $ub_1 = 2.618$ $ub_2 = 4.680$ $ub_3 = 48.34$ | 2.618 | ub_1 |
| Comb. 2 | $s_1 = 0.935$ $s_2 = 0.935$ $s_3 = 0.935$ $s_4 = 0.935$ $s_5 = 1.260$ | 1.348 | $ub_1 = 2.618$ $ub_2 = 2.008$ $ub_3 = 1.678$ | 1.678 | ub ₃ |
| Comb. 3 | $s_1 = 0.050$ $s_2 = 1.230$ $s_3 = 1.230$ $s_4 = 1.230$ $s_5 = 1.260$ | 25.20 | $ub_1 = 2.618$ $ub_2 = 2.008$ $ub_3 = 13.60$ | 2.008 | ub ₂ |
| Comb. 4 | $s_1 = 0.975$ $s_2 = 0.975$ $s_3 = 0.975$ $s_4 = 0.975$ $s_5 = 1.100$ | 1.128 | $ub_1 = 2.618$ $ub_2 = 2.000$ $ub_3 = 1.504$ | 1.504 | ub ₃ |
| Comb. 5 | $s_1 = 0.470$ $s_2 = 1.000$ $s_3 = 1.100$ $s_4 = 1.180$ $s_5 = 1.250$ | 2.660 | $ub_1 = 2.618$ $ub_2 = 2.000$ $ub_3 = 2.330$ | 2.000 | ub ₂ |

Table 2. Examples of the performance of the upper bounds for different parameter combinations in5-machine instances.

 $ub_1 = 2.618. \ ub_2 = 2 + (s_m - 1 - \frac{s_m}{m}) \mathbb{1}_{s_m > \frac{m}{m-1}}. \ ub_3 = \frac{1}{2} \left(\frac{s_m}{s_1} + s_m + 1 - \frac{s_m}{m} \right).$

6. Conclusions

This study presents two new parametrical upper bounds for the worst-case performance guarantee of Jump-Opt solutions for the problem of scheduling jobs in a uniformly related parallel machine environment to minimize total weight completion time, a recognized NP-hard combinatorial optimization problem.

The research focused on establishing the local optimality condition for the Jump neighborhood and on developing a mapping to represent a schedule within an inner product space, where the norm closely corresponds to the total weighted completion time of the schedule. The determined upper bounds establish the performance guarantee based on the parameters that describe an instance family, the number of machines, and the speed of the fastest and slowest machines.

The noteworthy findings of this study include the complementarity of the developed parametric upper bounds with the fixed performance guarantee of a generalization of the problem under study. Additionally, the new bounds outperformed the parametric upper bound previously reported in the literature.

Author Contributions: Conceptualization, F.T.M.; Methodology, F.T.M. and G.L.-N.; Validation, G.L.-N. and M.R.-M.; Formal analysis, M.R.-M.; Investigation, F.T.M.; Writing—original draft, F.T.M.; Writing—review & editing, G.L.-N. and M.R.-M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Universidad del Bío-Bío grant number 2060240 IF/R.

Data Availability Statement: There are no data used for the above study.

Acknowledgments: The authors would like to thank the editor and anonymous reviewers for their valuable comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Graham, R.L.; Lawler, E.L.; Lenstra, J.K.; Rinooy Kan, A.H.G. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Ann. Discret. Math.* **1979**, *5*, 287–326. [CrossRef]
- 2. Chen, H.; Guo, P.; Jimenez, J.; Dong, Z.S.; Cheng, W. Unrelated parallel machine photolithography scheduling problem with dual resource constraints. *IEEE Trans. Semicond. Manuf.* 2022, *36*, 100–112. [CrossRef]
- 3. Ji, B.; Xiao, X.; Yu, S.S.; Wu, G. A hybrid large neighborhood search method for minimizing makespan on unrelated parallel batch processing machines with incompatible job families. *Sustainability* **2023**, *15*, 3934. [CrossRef]
- 4. Siemiatkowski, M.S.; Deja, M. Planning optimised multi-tasking operations under the capability for parallel machining. *J. Manuf. Syst.* 2021, *61*, 632–645. [CrossRef]
- Berthier, A.; Yalaoui, A.; Chehade, H.; Yalaoui, F.; Amodeo, L.; Bouillot, C. Unrelated parallel machines scheduling with dependent setup times in textile industry. *Comput. Ind. Eng.* 2022, 174, 108736. [CrossRef]
- 6. Hidri, L.; Alqahtani, A.; Gazdar, A.; Ben Youssef, B. Green scheduling of identical parallel machines with release date, delivery time and no-idle machine constraints. *Sustainability* **2021**, *13*, 9277. [CrossRef]
- Vázquez-Serrano, J.I.; Cárdenas-Barrón, L.E.; Peimbert-García, R.E. Agent scheduling in unrelated parallel machines with sequence- and agent–machine–dependent setup time problem. *Mathematics* 2021, 9, 2955. [CrossRef]
- 8. Mahafzah, B.; Jabri, R.; Murad, O. Multithreaded scheduling for program segments based on chemical reaction optimizer. *Soft Comput.* **2021**, *25*, 2741–2766. [CrossRef]
- 9. Abu-Marrul, V.; Martinelli, R.; Hamacher, S.; Gribkovskaia, I. Matheuristics for a parallel machine scheduling problem with non-anticipatory family setup times: Application in the offshore oil and gas industry. *Comput. Oper. Res.* **2021**, *128*, 105162. [CrossRef]
- 10. Antunes, A.R.; Matos, M.A.; Rocha, A.M.A.C.; Costa, L.A.; Varela, L.R. A statistical comparison of metaheuristics for unrelated parallel machine scheduling problems with setup times. *Mathematics* **2022**, *10*, 2431. [CrossRef]
- 11. Durasević, M.; Jakobović, D. Heuristic and metaheuristic methods for the parallel unrelated machines scheduling problem: A survey. *Artif. Intell. Rev.* 2023, *56*, 3181–3289. [CrossRef]
- Komari Alaei, M.R.; Soysal, M.; Elmi, A.; Banaitis, A.; Banaitiene, N.; Rostamzadeh, R.; Javanmard, S. A Bender's algorithm of decomposition used for the parallel machine problem of robotic cell. *Mathematics* 2021, 9, 1730. [CrossRef]
- Módos, I.; Šucha, P.; Hanzálek, Z. On parallel dedicated machines scheduling under energy consumption limit. Comput. Ind. Eng. 2021, 159, 107209. [CrossRef]
- 14. Sterna, M. Late and early work scheduling: A survey. Omega-Int. J. Manag. Sci. 2021, 104, 102453. [CrossRef]
- 15. Xiao, Y.; Zheng, Y.; Yu, Y.; Zhang, L.; Lin, X.; Li, B. A branch and bound algorithm for a parallel machine scheduling problem in green manufacturing industry considering time cost and power consumption. *J. Clean Prod.* **2021**, *320*, 128867. [CrossRef]
- Fathollahi-Fard, A.M.; Ahmadi, A.; Goodarzian, F.; Cheikhrouhou, N. A bi-objective home healthcare routing and scheduling problem considering patients' satisfaction in a fuzzy environment. *Appl. Soft. Comput.* 2020, 93, 106385. [CrossRef]
- 17. Sepúlveda, I.A.; Aguayo, M.M.; De la Fuente, R.; Latorre-Núñez, G.; Obreque, C.; Orrego, C.V. Scheduling mobile dental clinics: A heuristic approach considering fairness among school districts. *Health Care Manag. Sci.* **2022**, 1–26. [CrossRef]
- 18. Çanakoğlu, E; Muter, İ. Identical parallel machine scheduling with discrete additional resource and an application in audit scheduling. *Int. J. Prod. Res.* 2021, *59*, 5321–5336. [CrossRef]
- 19. Rivera, G.; Porras, R.; Sanchez-Solis, J.P.; Florencia, R.; García, V. Outranking-based multi-objective PSO for scheduling unrelated parallel machines with a freight industry-oriented application. *Eng. Appl. Artif. Intell.* **2022**, *108*, 104556. [CrossRef]
- 20. Theophilus, O.; Dulebenets, M.A.; Pasha, J.; Lau, Y.Y.; Fathollahi-Fard, A.M.; Mazaheri, A. Truck scheduling optimization at a cold-chain cross-docking terminal with product perishability considerations. *Comput. Ind. Eng.* **2021**, *156*, 107240. [CrossRef]
- 21. Linfati, R.; Gatica, G.; Escobar, J.W. A mathematical model for scheduling and assignment of customers in hospital waste collection routes. *Appl. Sci.* 2021, *11*, 10557. [CrossRef]
- 22. Tian, G.; Fathollahi-Fard, A.M.; Ren, Y.; Li, Z.; Jiang, X. Multi-objective scheduling of priority-based rescue vehicles to extinguish forest fires using a multi-objective discrete gravitational search algorithm. *Inf. Sci.* **2022**, *608*, 578–596. [CrossRef]
- 23. Pasha, J.; Elmi, Z.; Purkayastha, S.; Fathollahi-Fard, A.M.; Ge, Y.E.; Lau, Y.Y.; Dulebenets, M.A. The drone scheduling problem: A systematic state-of-the-art review. *IEEE Trans. Intell. Transp. Syst.* **2022**, 23, 14224–14247. [CrossRef]
- 24. Pinedo, M.L. Scheduling: Theory, Algorithms, and Systems, 5th ed.; Springer: Berlin/Heidelberg, Germany, 2016; pp. 13–21.
- 25. Conway, R.W.; Maxwell, W.L.; Miller, L.W. Theory of Scheduling; Addison-Wesley: Boston, MA, USA, 1967; pp. 74–79.
- 26. Horowitz, E.; Sahni, S. Exact and approximate algorithms for scheduling nonidentical processors. J. ACM 1976, 23, 317–327. [CrossRef]
- 27. Horn, W.A. Minimizing average flow time with parallel machines. Oper. Res. 1973, 21, 846–847. [CrossRef]
- 28. Garey, M.; Johnson, D. Computers and Intractability: A Guide to the Theory of NP-Completeness; WH Freeman and Co.: San Francisco, CA, USA, 1979.
- 29. Bruno, J.; Coffman, E.G., Jr.; Sethi, R. Scheduling independent tasks to reduce mean finishing time. *Commun. ACM* **1974**, 17, 382–387. [CrossRef]

- 30. Lenstra, J.K.; Rinooy Kan, A.H.G.; Brucker, P. Complexity of machine scheduling problems. *Ann. Discret. Math.* **1977**, *1*, 343–362. [CrossRef]
- Garey, M.; Johnson, D. Strong NP-Completeness results: Motivation, examples, and implications. J. ACM 1978, 25, 499–508. [CrossRef]
- Lawler, E.L.; Lenstra, J.K.; Rinnooy Kan, A.H.G.; Shmoys, D.B. Chapter 9 Sequencing and scheduling: Algorithms and complexity. In *Logistics of Production and Inventory, Handbooks in Operations Research and Management Science*; Graves, S.C., Rinnooy Kan, A.H.G., Zipkin, P.H., Eds.; Elsevier: Amsterdam, The Netherlands, 1993; Volume 4, pp. 445–522. [CrossRef]
- Epstein, L.; Sgall, J. Approximation schemes for scheduling on uniformly related and identical parallel machines. *Algorithmica* 2004, *39*, 43–57. [CrossRef]
- 34. Skutella, M.; Woeginger, G.J. A PTAS for minimizing the total weighted completion time on identical parallel machines. *Math. Oper. Res.* **2000**, *25*, 63–75. [CrossRef]
- Bansal, N.; Srinivasan, A.; Svensson, O. Lift-and-round to improve weighted completion time on unrelated machines. In Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing, Cambridge, MA, USA, 19–21 June 2016; pp. 156–167. [CrossRef]
- Li, S. Scheduling to minimize total weighted completion time via time-indexed linear programming relaxations. *SIAM J. Comput.* 2020, 49, 409–440. [CrossRef]
- Williamson, D.P.; Shmoys, D.B. *The Design of Approximation Algorithms*; Cambridge University Press: Cambridge, UK, 2011; pp. 14–15. Available online: https://www.designofapproxalgs.com/index.php (accessed on 8 November 2023).
- Angel, E. A survey of approximation results for local search algorithms. In *Efficient Approximation and Online Algorithms. Lecture Notes in Computer Science*; Bampis, E., Jansen, K., Kenyon, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; Volume 3484, pp. 30–73. [CrossRef]
- Michiels, W.; Aarts, E.; Korst, J. Theoretical Aspects of Local Search; Springer Science & Business Media: Berlin, Germany, 2007. [CrossRef]
- Ahuja, R.K.; Ergun, Ö.; Orlin, J.B.; Punnen, A.P. A survey of very large-scale neighborhood search techniques. *Discret. Appl. Math.* 2002, 123, 75–102. [CrossRef]
- 41. Muñoz, F.T.; Pinochet, A.A. Performance guarantee of the jump neighborhood for scheduling jobs on uniformly related machines. *Rairo-Oper. Res.* 2022, *56*, 1079–1088. [CrossRef]
- 42. Correa, J.R.; Muñoz, F.T. Performance guarantees of local search for minsum scheduling problems. *Math. Program.* 2022, 191, 847–869. [CrossRef]
- 43. Brueggemann, T.; Hurink, J.L.; Kern, W. Quality of move-optimal schedules for minimizing total weighted completion time. *Oper. Res. Lett.* **2006**, *34*, 583–590. [CrossRef]
- 44. Finn, G.; Horowitz, E. A linear time approximation algorithm for multiprocessor scheduling. Bit 1979, 19, 312–320. [CrossRef]
- 45. Schuurman, P.; Vredeveld, T. Performance guarantees of local search for multiprocessor scheduling. *INFORMS J. Comput.* **2007**, 19, 52–63. [CrossRef]
- 46. Cho, Y.; Sahni, S. Bounds for list schedules on uniform processors. SIAM J. Comput. 1980, 9, 91–103. [CrossRef]
- Recalde, D.; Rutten, C.; Schuurman, P.; Vredeveld, T. Local search performance guarantees for restricted related parallel machine scheduling. In *LATIN 2010: Theoretical Informatics. Lecture Notes in Computer Science*; López-Ortiz, A., Ed.; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6034, pp. 108–119. [CrossRef]
- Rutten, C.; Recalde, D.; Schuurman, P.; Vredeveld, T. Performance guarantees of jump neighborhoods on restricted related parallel machines. *Oper. Res. Lett.* 2012, 40, 287–291. [CrossRef]
- 49. Brueggemann, T.; Hurink, J.L.; Vredeveld, T.; Woeginger, G.J. Exponential size neighborhoods for makespan minimization scheduling. *Nav. Res. Logist.* 2011, 58, 795–803. [CrossRef]
- Frangioni, A.; Necciari, E.; Scutellà, M.G. A multi-exchange neighborhood for minimum makespan parallel machine scheduling problems. J. Comb. Optim. 2004, 8, 195–220. [:JOCO.0000031420.05971.29CrossRef] [CrossRef]
- Brucker, P.; Hurink, J.; Werner, F. Improving local search heuristics for some scheduling problems. Part II. *Discret. Appl. Math.* 1997, 72, 47–69. [CrossRef]
- 52. Eastman, W.L.; Even, S.; Isaacs, I.M. Bounds for the optimal scheduling of n jobs on m processors. *Manage. Sci.* **1964**, *11*, 268–279. [CrossRef]
- 53. Smith, W.E. Various optimizers for single-stage production. *Nav. Res. Logist. Q.* **1956**, *3*, 59–66. [CrossRef]
- 54. Cole, R.; Correa, J.R.; Gkatzelis, V.; Mirrokni, V.; Olver, N. Decentralized utilitarian mechanisms for scheduling games. *Games Econ. Behav.* 2015, *92*, 306–326. [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.