Available online at : http://josi.ft.unand.ac.id/

Jurnal Optimasi Sistem Industri

| ISSN (Print) 2088-4842 | ISSN (Online) 2442-8795 |

Research Article

Total Tardiness Minimization in a Single-Machine with Periodical Resource Constraints

Bruno de Athayde Prata [1], Levi Ribeiro de Abreu [2], and Marcelo Seido Nagano [2]

[1] *Federal University of Ceara, Fortaleza, Brazil.*
[2] *University of Sao Paulo, São Carlos, Brazil*

ARTICLE INFORMATION

Received : August 8, 2022
Revised : November 29, 2022
Available online : November 30, 2022

KEYWORDS

Production sequencing, Combinatorial Optimization, Matheuristics, Resource Consumption.

CORRESPONDENCE

Phone: +55 (85)33669882
E-mail: baprata@ufc.br

A B S T R A C T

In this paper we introduce a variant of the single machine considering resource restriction per period. The objective function to be minimized is the total tardiness. We proposed an integer linear programming modeling based on a bin packing formulation. In view of the NP-hardness of the introduced variant, heuristic algorithms are required to find high-quality solutions within an admissible computation times. In this sense, we present a new hybrid matheuristic called Relax-and-Fix with Variable Fixing Search (RFVFS). This innovative solution approach combines the relax-and-fix algorithm and a strategy for the fixationof decision variables based on the concept of the variable neighborhood search metaheuristic. As statistical indicators to evaluate the solution procedures under comparison, we employ the Average Relative Deviation Index (ARDI) and the Success Rate (SR). We performed extensive computational experimentation with a testbed composed by 450 proposed test problems. Considering the results for the number of jobs, the RFVFS returned ARDI and SR values of 35.6% and 41.3%, respectively. Our proposal outperformed the best solution approach availablefor a closely-related problem with statistical significance.

INTRODUCTION

The single-machine sequencing problem has been studied by operational research practitioners in the last decades in vew of the several applications of this production environment in real-world scenarios. Although to process the jobs a single machine environment is used, this production environment usually belongs to the NP-hard class for several objective functions. In the last few decades, several researchers have paid attention to this production environment, and new features of this problem have been introduced. Nowadays, because of the environmental issues related to the planning process in industries, sustainable manufacturing or green scheduling is a relevant research branch. Usually, in a typical production plant, the planners manage several resources with distinct levels of scarcity.

In the literature of deterministic scheduling, resource consumption is usually ignoredin the planning process. However, several examples in practice illustrate that resourceconsumption can be relevant in this context. In recent years, resource consumption constraints have been widely studied in production scheduling problems [3, 9]; nevertheless, the consideration of resource consumption per period is quite limited in theavailable literature [25]. Much attention has been paid to single-machine scheduling problems in the last decades [13]; however, the consideration of resource consumption in this class of problems is rather limited in comparison with other production environments.

Another research line is the development of efficient solution procedures to solve combinatorial optimization problems. In recent decades, a crescent interest in heuristics and metaheuristic algorithms was observed, given their robustness and modularity. Recently, with the development of commercial solvers of mixed-integer linear programming models, several operational research practioners have developed hybrid solution approaches called matheuristics. In such solution methods, heuristic algorithms are included in the mathematical programming models, aiming to reduce the computational time required to solve medium-sized and large-sized test instances.

In this paper, we address a single-machine scheduling environment with periodical resource constraints to minimize the total tardiness. To the best of our knowledge, this variant is not

DOI: 10.25077/josi.v21.n2.p97-105.2022

reported yet in the revised literature. The studied order scheduling variant was based in many real-world scenarios, like precast production [24]. We can consider the production of precast beams, with distinct lengths and due dates, which are produced in a single form. In each productive period, we can observe the consumption of several resources, such as time, labor, or materials, which are scarce.

This paper addresses a single-machine scheduling environment with periodical resource constraints and total tardiness minimization. The new single machine problem has processing time and resource consumption parameters. In addition, each available production period has a time and resource capacity that can be allocated to jobs. Because of such new features, the problem under study is closely related to the one-dimensional bin packing problem.

The main contributions of this paper are the following: (1) we introduce a new variant for the single-machine environment relevant to practice; (2) we develop a mixed-integer linear programming (MILP) modeling for the addressed variant; (3) we propose an innovative hybrid matheuristic that uses a random and iterative process to fix decision variables; and (4) we present a testbed with 450 randomly generated test instances.

A production scheduling problem under proposition is characterized by a single-machine layout in which the jobs have different processing times, resource consumption, and due dates. Furthermore, each planning period has two constraints related to processing times and resource consumption. Since the problem under study is not reported yet, we present some contributions to problems with resource consumption. Several contributions to resource consumption have been addressed to the parallel-machine environment. Thereby, we present here several contributions in this production environment, which are related to the problem under study.

Ventura and Kim [31] presented a parallel machines scheduling layout with resource constraints. The performance measure is the minimization of earliness and tardiness penalties. Two heuristics were proposed: the first one generates an initial solution, and the second one changes an unfeasible solution to a feasible solution. Edis and Ozkarahan [8] a variant of the parallel machines scheduling problem with addressed a resource constraints that arises in the injection-molding processes of electrical appliance plants. Since the proposed MILP model failed to solve the problem efficiently, two procedures to decompose the original problem into subproblems were presented. Ji et al. [14] and Yeh et al. [36] studied the resource consumption variant of uniform parallel-machine scheduling problem. Ji et al. [14] developed a constructive heuristic and a particle swarm optimization (PSO) metaheuristic to tackle the problem. On the other hand, Yeh et al. [36] presented a genetic algorithm (GA) as well as two PSO-based metaheuristics as solution procedures. Afzalirad and Rezaejan [1] addressed an unrelated parallel machine scheduling problem considering the following characteristics: resource constraints, machine eligibility, release dates, sequence-dependent setup times, and precedence constraints. Two evolutionary algorithms are developed as solution approaches. Afzalirad and Shafipour [2] introduced an unrelated parallel machine environment with eligibility constraints and multiple resources. Besides, two GA-based metaheuristics are presented as solution procedures. Fanjul-Peyro et al. [10] considered a parallel-machine variant in a production window with limited and fixed resources. Besides, job processing requires the consumption of scarce resources. Three matheuristics strategies were presented to tackle the problem, all of them based on the reduction of decision variables in the original model. Villa et al. [32] considered an unrelated parallel machine environment with one scarce additional resource. Also, several constructive heuristics and local search procedures were presented.

We can observe that contributions considering resource consumption constraint in the single-machine scheduling problem are still limited, in comparison with the contributions reported to parallel-machine problems. Wang et al. [33] introduced a single-machine variant in which processing times are determined by means of a procedure of resource usage. The performance measure is the reduction of the total amount of used resources subject to a constrained total weighted flow time. A branch-and-bound (B&B) algorithm was proposed as solution procedure. Wang and Wang [34] studied the problem of processing times with deterioration in a single-machine environment. The performance measure of the problem is the aggregation of the following objectives: makespan, total completion time, the difference between completion time and resource usage, and a cost function.

Furthermore, the authors demonstrate that the problem is solvable in polynomial time. Zhu et al. [37] studied a single-machine scheduling problem in which a resource allocation function determines the processing time of a given job. Also, these authors showed that the problem can be solved in polynomial time. Wu and Cheng [35] presented a single-machine environment with resource constraints, which is a non-linear combinatorial optimization problem that appears in cloud computing applications. In specific situations, this variant is solvable in polynomial time. Herr and Goel [13] considered a single-machine scheduling problem with family setups and resource constraints to minimize total tardiness. Two MILP models were described for two variants of the problem under study observed in the stell industry. Also, an iterated local search (ILS) metaheuristic was developed to solve large-sized problems. For this variant, Pinheiro and Arroyo [22] presented an iterated greedy (IG) metaheuristic, which outperforms the ILS algorithm proposed by Herr and Goel [13]. Karky and Shabtay [15] approached a single-machine scheduling problem to minimize resource consumption where the job processing times and due dates are decision variables. Since this variant is NP-hard, approximation algorithms were presented. Shabtay [29] addressed a single-machine scheduling problem with a machine-unavailability period. In this variant, processing times are resource-dependent. As such variant is an NP-hard problem, some pseudo-polynomial time and approximation algorithms were developed. Concerning the periodical resource constraints, Prata et al. [25] introduced the single-machine scheduling problem with periodical resource constraints. As solution approaches, a MILP formulation was presented, as well as [18], local search algorithm, approximation heuristic approaches, and a matheuristic that hybridizes a size-reduction with simulated annealing were proposed.

Based on the literature review, we can emphasize the following

 DOI: 10.25077/josi.v21.n2.p97-105.2022

research gaps. The literature review found that the single-machine scheduling problem with resource consumption constraints and total tardiness minimization has not yet been studied. Therefore, it is a gap to be filled in the present study. Since this production environment was not studied yet, there are not test instances available in the literature. In this context, a new testbed must be provided. Given the distinct characteristics of the variant under proposition, innovative solution approaches could be developed.
The remaining sections of this paper are described below. Section 2 defines the variant under proposition and the proposed solution approach. Section 3 presents the experimental setup, the computational experimentation, and the discussion of the results. Section 4 provides the main findings and suggestions for future studies.

METHODS

Although mathematical programming formulation can be a limited solution procedure to solve NP-hard problems, the modeling process contributes to the problem definition and comprehension. In addition, for small-sized test instances, integer linear programming models can return high-quality solutions and, in some particular cases, global optimal solutions.

Distinct MILP models have been presented for the single-machine scheduling problems in the last few decades. We can emphasize models with completion time variables [16], time-indexed variables [30], or positional variables [7]. Since the variant under proposition has a close connection with cutting and packing problems [25], we develop here a formulation based on the one-dimensional bin packing problem, which is a well-known NP-hard problem [20].

Let n be the number of jobs with an associated processing time pj, a resource consumption rj, and a due date dj to be processed in a single machine. Each planning period has P time units, and a periodical constraint of resources limited by R. Each job j has a completion time Cj and a tardiness Tj, which can be calculated as in Equation (1):

$$T_j = \max\{0, C_j - d_j\} \quad (1)$$

Traditionally, production scheduling problems that consider tardiness-related objectives consider a linear calculation of the tardiness based on the difference between the completion time of the job and its due date. In our modeling, we calculate *Cj* as the planning period *i* in which the job *j* is produced. Thereby, we do not create a decision variable for *Cj* since it can be determined in terms of *i*. In this context, the production environment under study presents a relation with the one-dimensional bin packing problem.

Here, we present a MILP model for the variant under proposition.

$$min z = \sum_{j=1}^{n} T_i \quad (2)$$

Subject to:

$$\sum_{i=1}^{m} x_{ij} = 1, \qquad \forall j \quad (3)$$

$$\sum_{j=1}^{n} p_j x_{ij} \leqslant P y_i, \qquad \forall i \quad (4)$$

$$\sum_{j=1}^{n} r_j x_{ij} \leqslant R y_i, \qquad \forall i \quad (5)$$

$$T_j \geq i x_{ij} - d_j, \qquad \forall i,j \quad (6)$$

$$T_j \geq 0, \qquad \forall j \quad (7)$$

$$y_i \in \{0,1\}, \qquad \forall i \quad (8)$$

$$x_{ij} \in \{0,1\}, \qquad \forall i,j \quad (9)$$

Equation (2) illustrates the minimization of total tardiness as an objective function. Constraints (3) force that all jobs are allocated in a single period. Constraints (4) and (5) ensure that each period respects the time and resource consumption capacity. Constraint (6) illustrates how tardiness is calculated for each job. The sets of constraints (7), (8), and (9) represent the domain of the decision variables. The developed formulation has $m(n + 1)$ binary decision variables, n continuous decision variables, and $2n(1 - m) + 3m$ integer linear constraints.

The estimation of the number of planning periods plays a key issue in the efficiency of MILP models for scheduling problems with multiple periods [23]. The higher the number of planning periods, the more decision variables in the model. Since not all of these decision variables will be selected in the optimal solution, a more accurate estimate of the number of planning periods can improve the resolution process. However, if the number of periods is not sufficient to process all the jobs, the model would not have a feasible solution.

We adopt the same estimation presented by Prata et al. [25], considering the Next-Fit (NF) algorithm [28]. NFP is the amount of minimum periods considering only the time constraint and NFR, is the amount of periods considering only the resource consumption constraint, the number of periods upper bound can be obtained by:

$$UB = \max\{NF_P, NF_R\} \quad (10)$$

The lower bound for the number of planning periods can be determined using Equation (11) [25]:

$$LB = round\{max\{\frac{\sum_{j=1}^{n} p_j}{T}, \frac{\sum_{j=1}^{n} r_j}{R}\}\} \quad (11)$$

Let us consider a toy model with $p = \{2, 1, 5, 4, 3\}$, $r = \{3, 3, 4, 1, 1\}$, $P = 5$, $R = 4$, $d = \{3, 2, 1, 2, 2\}$, and $UB = 4$ – calculated as in Equation (10). Based on this data, the optimal solution is the sequence $\Pi = 3, 2, 4, 1, 5$, with a total tardiness of 1 time unit. In this solution, the completion time and tardiness vectors are given by $C = \{3, 2, 1, 2, 3\}$ and $T = \{0, 0, 0, 0, 1\}$. Figure 1 illustrates this solution, in which ρ in the resource consumption per period.

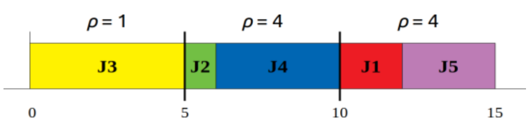


Figure 1. Gantt Chart for the Illustrative Example

Concerning the proposed variant, the following main comments can be emphasized. First, even for a single-machine problem, slacks in each production period can appear if the resources are not fully used in a given planning period. For this reason, the variant is closely related to the one-dimensional bin packing problem. If the empty spaces in each planning period are reduced, the solution generated tends to present lower total tardiness. Second, the bin packing problem presents two constraints: the first one is related to the planning periods, and the second one is related to resource consumption. In this context, the complexity of the scheduling problem under study is higher than a standard one-dimensional bin packing problem. Finally, the tardiness for each job is calculated differently to other due-date based scheduling problems. Usually, the linear tardiness for a given job is established comparing the completion time of this job with its respective due date. In our variant, we determine the tardiness with respect to the planning period in which the job is processed.

Metaheuristics present as their main advantages the modularity and robustness for solving combinatorial optimation problems. Nevertheless, such algorithms do not provide a notion of distance for the solution returned and the optimal solution. On the other hand, mathematical programming approaches present a gap of the current solution to the best-so-far lower bound obtained in the search process. However, the computational cost to solve larged-sized or even medium-sized test instances can be prohibitive in certain circumstances. Currently, several authors have applied the hybridization of heuristics and integer linear programming to solve production scheduling problems [4, 6, 12, 18, 23, 26].

The number of decision variables plays a key role in the resolution process of an integer programming model in a commercial solver. Usually, robust solvers present a pre-processing stage in which some decision variables and problem constraints are removed, aiming to reduce the computational times to solve a given combinatorial optimization problem. Nevertheless, such preliminary processing also can consume a considerable computational time, mainly for large-sized test problems. In addition, for a given test instance, several decision variables of an integer programming model can be fixed as zeros because of the low possibility of such variables appearing in the global optimal solution. For example, the decision variable associated with the parameter with the largest cost probably will not be selected in the global optimal solution. In this context, we propose an innovative matheuristic that explores in each iteration a distinct structure for variable fixing.

The variable neighborhood search [21] is a recognized metaheuristic that investigates several heterogeneous neighborhoods of the incumbent solution. The algorithm changes the neighborhood to a new one considering the improvement in the value of the objective function. Recently, some researchers have proposed hybrid matheuristics combining mathematical programming and the VNS framework for other combinatorial optimization problems such as nurse scheduling [5, 27] and bin packing problems [17].

In accordance with the experience of the author, the fixation process of decision variables is a complex task. Although one can deterministically determine the decision variables to be fixed based on the parameters of a given test instance, this way can lead to a deterioration in the value of the objective function found. As in the case of heuristic algorithms, the use o deterministic procedures can lead the search process to locally optimal solutions. In this context, probabilistic operators can propitiate that the search process scape from local optimal.

In the developed solution procedure, we explore several fixations during the search process. Initially, we employ the information of the linear relaxation to guide the fixation of the decision variables. Based on preliminary experiments, we could observe that if a decision variable has a null value in the relaxed solution, it will probably have the null value also the solution with integer decision variables. The first step of our algorithm is to solve the linear relaxation and store the results in the matrix x^{relax}.

Considering the values of x^{relax}, we interactively fix with zero some decision variables in the mixed-integer linear programming model based on a given probability. In the first iteration of this process, the search process in the mixed-integer model begins with an empty solution. For all i and j, if $x^{relax} = 0$, a random number in the interval [0, 1] is then generated. If this random value is higher than the probability *prob*, the integer decision variable x_{ij} is fixed as zero. After a partial time limit, which is given by $t_{partial} = t_{limit}/n_{starts}$, the better integer solution found is stored in x^{warm} and used as a warm start for the next iteration. This process is repeated until the specified time limit is not exceeded.

Based on the above, our proposed matheuristic called Relax-and-Fix with Variable Fixing Search (RFVFS) is summarized as in Algorithm 1. As inputs of the proposed matheuristic, we have an instance of the problem under study, as defined in Section 3, as well as the two parameters of the algorithm (n_{starts} and *prob*). As outputs of the proposed algorithm, we have A sequence π and a total tardiness *TT*.

Algorithm 1: Relax-and-Fix with Variable Fixing Search (RFVFS)

Input: p_j, r_j, d_j, P, R, t_{limit}, n_{starts}, *prob*
Output: A schedule (π) and a total tardiness TT
Solve the linear relaxation of the problem – Equations (2)-(9) – and store the values of decisionvariable x_{ij} in x^{relax}
$k \leftarrow 1$
while $k \leq n_{starts}$ **do**
if $k = 1$ **then**
Solve the problem – Equations (2)-(9) – and store the values of decision variable x_{ij} in x^{warm}
$k \leftarrow k+1$
else
Use the values of x^{warm} obtained in the previous iteration as a warm start for the current iteration.
Solve the problem – Equations (2)-(9) – and store the values of decision variable x_{ij} in x^{warm}
$k \leftarrow k+1$

RESULTS AND DISCUSSION

We employ the Relative Deviation Index (*RDI*) [11] to evalaute the methods under comparison. Let H be a set of solution procedures, the *RDI* found for the method s ⍰ H when used to the test instance t is determined as in Equation (12).

DOI: 10.25077/josi.v21.n2.p97-105.2022

$$I_{st} = \begin{cases} 0, & \text{if } \min_{h\in H} T_{ht} = \max_{h\in H} T_{ht}, \\ \dfrac{T_{st} - \min_{h\in H} T_{ht}}{\max_{h\in H} T_{ht} - \min_{h\in H} T_{ht}} \cdot 100, & \text{otherwise.} \end{cases} \quad (12)$$

where *Tst* is the tardiness value returned by solution procedure s in the test instance *t*. In our study, $\min_{h\in H}$ Tht is the best solution found among the methods under comparison. We used the average RDI (ARDI) for each set of instances as aggregate performance measure.

We also used the success rate (SR) as another performance measure. It is the ratio of the times a method finds the best solution divided by the total number of tested instances. Equation (13) expresses the calculation of the SR indicator.

$$SR = \frac{n_{BEST}}{n_{INST}} \times 100 \quad (13)$$

where n_{BEST} is the number of test instances in which the solution approach found the best objective function value and n_{INST} is the number of evaluated test instances.

Since the variant under study has not been previously addressed, we need to generate a new set of test instances. Additional information not available in the traditional variants of the single-machine scheduling problem becomes necessary. Aside from the generation of the processing times, in our variant, we also need to generate resource consumption for all jobs, constraints for the planning period, and resource consumption per period.

The production scheduling problem under study has a peculiar characteristic where the sequencing process considers two one-dimensional bin packing problems. The first one is related to the processing times constrained to a maximal duration of the planning period. The second one is related to the resource consumption for each job and the maximal amount of resources for each planning period. After several preliminary computational experiments, we could observe that processing times and resource consumption for each job generated with a uniform distribution [50, 100] generates difficult test instances. The associated knapsack constraints also play a key role in the difficult of a given test problem. After several previous simulations, we could observe that such constraints could be set with a value of 200.

We consider the following parameters for the testbed generation: the number of jobs (*n*), the tardiness factor (*TR*), and the due date range (*RDD*), as described in Table 1. We defined the values of these parameters taking the previous literature into account. The total number of instance classes is given by 5 (*n*) × 3 (*TR*) × 3 (*RDD*) = 45. Aiming to reduce the sampling error, we generated 10 test instances for each class, resulting in 450 test problems.

Although metaheuristics are robust solution procedures to solve hard combinatorial optimization problems, one of their main advantages is the need for the calibration of several parameters. In this paper, we developed a matheuristic with a low dependency on parameters. In the design of the algorithm, we perceived a tradeoff between diversification and intensification related to the number of restarts. In addition, after several preliminary experiments, we can show that the fixation of the decision variables presents a relevant impact on the quality of the solutions found. If we fix a given number of decision variables less than or equal to ten percent, the fixation process does not present a significant benefit. On the other hand, if we fixed more than fifty percent of the number of decision variables, we could observe a substantial loss in the quality of the solutions returned by the proposed matheuristic. In addition, after several simulations, we could not observe a significant discrepancy in the results obtained in a range of fifty percent and ninety percent of fixation for the decision variables.

Table 1. Parameters Employed in the Test Instance Generation

Parameters	**Levels**
Number of jobs	$n \in \{50, 100, 250, 500, 1000\}$
Processing time distribution	[50, 100]
Resource consumption distribution	[50, 100]
Maximum duration of the planning periods	200
Maximum amount of resource for each planning period	200
Tardiness factor	[0.2, 0.5 ,0.8]
Due date ranges	[0.2, 0.5, 0.8]

Although metaheuristics are robust solution procedures to solve hard combinatorial optimization problems, one of their main advantages is the need for the calibration of several parameters. In this paper, we developed a matheuristic with a low dependency on parameters. In the design of the algorithm, we perceived a tradeoff between diversification and intensification related to the number of restarts. In addition, after several preliminary experiments, we can show that the fixation of the decision variables presents a relevant impact on the quality of the solutions found. If we fix a given number of decision variables less than or equal to ten percent, the fixation process does not present a significant benefit. On the other hand, if we fixed more than fifty percent of the number of decision variables, we could observe a substantial loss in the quality of the solutions returned by the proposed matheuristic. In addition, after several simulations, we could not observe a significant discrepancy in the results obtained in a range of fifty percent and ninety percent of fixation for the decision variables.

Our proposed matheuristic presents two parameters: the number of starts *(n_{starts}*) and the probability of fixing a decision variable (*prob*). Concerning the number of starts, we could observe a trade-off between diversification and intensification. In view of the adopted time limit (t_{limit} = *n* seconds), with a number of starts higher than 3 the matheuristic presented difficulty in finding high-quality solutions. Thus, we used n_{starts} = 3 in our proposal. With respect to the probability to fix decision variables, we can observe that probabilities in the range of 0.5 and 0.9 conducted to good results. In this sense, we used *prob* = 0.8 in our proposed solution procedure.

Aiming to evaluate the proposed mixed-integer linear programming and the proposed Relax-and-Fix with Variable Fixing Search, we perform computational experimentats with the 450 test instances generated. All the evaluated methods were implemented in Julia programming language (https://julialang.org/). The Integrated Development Environment used was the VSC (Visual Studio Code - https://code.visualstudio.com/).

For the MILP model and the matheuristics, we used the commercial solver Gurobi (https://www.gurobi.com/) version 9.0.3 with JuMP library (https://www.juliaopt.org/JuMP.jl/stable/) [19]. Weperformed the tests on a PC with Intel Core 2 Duo CPU 3.00GHz and 4GB memory, with the Windows 10 operating system.

Regarding solution procedures evaluated in the proposed test instances, the following approaches have been compared:

- The MILP model, as in Equations (2)-(9).
- The matheuristic Size Reduction with Simulated Annealing (SRSA), proposed by Prata et al. [25]. We selected this solution approach since it is developed for the most closely related problem identified in the literature review. We adapted this algorithm for the total tardiness objective.
- The matheuristic Relax-and-Fix with Variable Fixing Search (RFVFS) (our proposed solution approach).

Although there are several approximate algorithms for the single-machine scheduling problem with total tardiness minimization, in our view, a comparison with such solution approaches would not be fair. In the problem under study, we have resource consumption per period, and the total tardiness is determined using bin packing constraints. Based on the revised literature, the closely-related solution approach is the SRSA matheuristic developed by Prata et al. [25]. Although the above-mentioned solution procedure has been developed for the makespan minimization, the associated bin packing constraints are quite similar.

For all the methods under comparison, we use a time limit t_{limit} = *n* seconds. Besides, since the SRSA and RFVS are stochastic algorithms, we run each algorithm five times and report the average values. Table 2 describes the results of average RDI and SR values considering the number of jobs. In resume, the following analysis can be point out:

- The MILP model presented a similar performance for the test instances with 50, 100, and 250 jobs. However, for the test instances with 500 and 1000 jobs, there was a worsening in the RDI and SR values. For the test problems with 1000 jobs, the MILP model could not find any better solution. In addition, we can also observe that the MILP model returned competitive results for the test instances with 50 and 250 jobs.
- The SASR matheuristic presented a worse performance in the test instances with 50 and 100 jobs. For the test problems with 100 jobs, the SRSA algorithm could not find any better solution. For the test problems with 250 jobs, its performance was the best in comparison with the MILP model and RFVFS. For the test instances with 500 jobs, the SRSA and RFVFS algorithm presented a similar behavior.
- The RFVFS presented better results for the test instances with 50, 100, and 1000 jobs. For the test instances with 250, the RFVFS could not find any better solution. On the other hand, for the test instances with 1000 jobs, this solution procedure could find the best solution for all the evaluated problems.

Considering the average results for *n* values, we can observe that the MILP model and the SASR presented outputs of the same magnitude (respectively, 55.5% and 51.2% for the average RDI values; 30.4% and 29.3% for the SR values). On the other hand, the proposed RFVFS matheuristc returned better results for both indicators (respectively, 35.6% for the average RDI values and 41.3% for the SR values). This difference points out the superiority of the proposed solution approach when compared with the MILP model and the SASR matheuristic.

Table 3 illustrates the results of average RDI and SR values with respect to *TF* and *RDD*. In resume, the following analysis can be point out:

- The RFVFS presented an average RDI value of around 35% for all combinations of *TF* and *RDD* except for the test instances with *TF* = 0.2 and *RDD* = 0.2, in which the average RDI value was of 19.3%. This indicates that the proposed solution approach presented a stable behavior.
- In the same way, the RFVFS presented a stable value of SR of around 40% for all combinations of *TF* and *RDD* except for the test instances with *TF* = 0.2 and *RDD* = 0.2, in which the average RDI value was of 66.7%. This result reinforces the tendency of stability of the proposed solution approach.

Considering that we have evaluated three distinct solution approaches, we must determine if the difference between the results found is statistically significant. In this context, we performed a statistical analysis based on two steps. In the first one, we used the statistical test Analysis of Variance (ANOVA) to evaluate if there is a significant difference in the average relative percentage deviation for the solution approaches under comparison. The second one is Tukey's range test, in which a pairwise comparison among the evaluated methods is performed.

Figure 2 presents the boxplots for average relative deviation index values. In addition, we accomplish an ANOVA test to determine if

Table 2. Results for n values

	RDI			**SR**		
n	MILP	SASR	RFVFS	MILP	SASR	RFVFS
50	34.4	72	30.0	53.3	18.9	30
100	34.3	98.5	10.6	37.8	0	63.3
250	33.9	22.5	95.0	48.9	53.3	0
500	82.3	13.2	42.4	12.2	74.4	13.3
1000	92.6	49.6	0.0	0.0	0.0	100
Average	55.5	51.2	35.6	30.4	29.3	41.3

Table 3. Results for *TF* and *RDD* values.

		RDI			**SR**		
TF	**RDD**	**MILP**	**SASR**	**RFVFS**	**MILP**	**SASR**	**RFVFS**
0.2	0.2	70.1	52.4	19.3	18.9	14.4	66.7
0.2	0.5	65.3	42.6	33.6	26.0	32.0	42.0
0.2	0.8	50.4	52.5	39.5	32.0	28.0	40.0
0.5	0.2	57.4	50.2	34.7	28.0	34.0	40.0
0.5	0.5	57.3	51.1	37.7	30.0	28.0	44.0
0.5	0.8	65.5	47.8	33.1	18.0	40.0	42.0
0.8	0.2	47.6	53.5	36.1	38.0	26.0	38.0
0.8	0.5	53.6	53.0	33.9	36.0	26.0	40.0
0.8	0.8	51.5	55.3	36.9	32.0	24.0	46.0
Average		57.6	50.9	33.9	28.8	28.0	44.3

 DOI: 10.25077/josi.v21.n2.p97-105.2022

the difference between the ARPD values found is significant. Since the ANOVA returns a statistic 26.8, a value much higher than the critical value 4.39, the difference among the methods under comparison is statistically significant. After that, Figure 3 illustrates the Tukey's confidence intervals for ARDI values. We can observe that our proposed matheuristic outperform all other methods under comparison.

Based on the statistical analysis, the following comments can be addressed. First, the performed comparison is fair since we run all the methods under comparison in the same hardware, coded in the same programming language, and using the same time limit as the stop criterion. Second, based on Tukey's range test, the MILP model and the SRSA matheuristic presented results without a statistically significant difference. Thereby, we verified a draw between both solution approaches. Third, given the pairwise comparison performed, we can emphasize that the proposed Relax-and-Fix with Variable Fixing Search returned better results than the MILP model and the SRSA algorithm

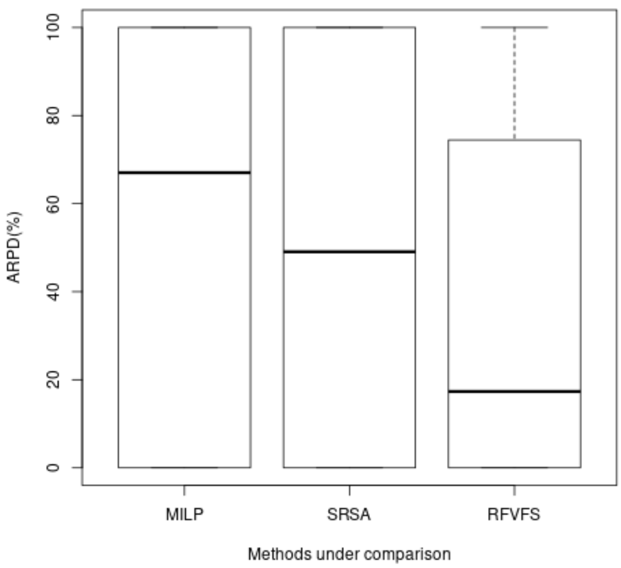


Figure 2. Boxplot for ARDI Values fortThe Methods under Comparison

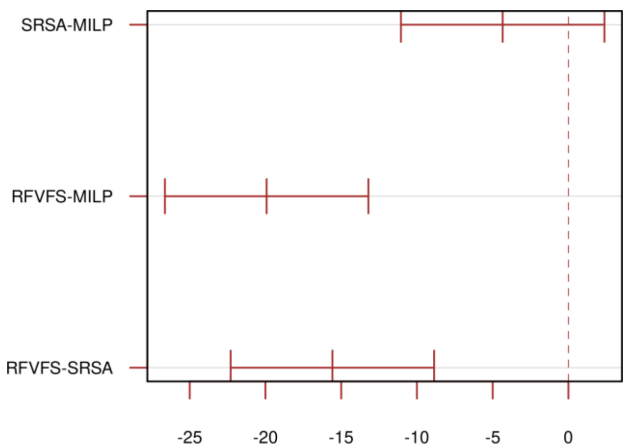


Figure 3. Tukey's Confidence Intervals for ARDI Values

CONCLUSIONS

In this paper, we introduced a new variant of the single-machine scheduling problem, taking periodical resource constraints into consideration. The performance measure is total tardiness minimization. A MILP formulation is proposed for the problem under study. Furthermore, some problem properties have been addressed. As the solution procedure approach, we presented a hybrid matheuristic named Relax-and-Fix with Variable Fixing Search (RFVFS). The developed algorithm iteratively fixates several decision variables using a variable neighbourhood search framework. This probabilistic procedure was able to find promising results.

Since the problem under study is not reported yet, we proposed a new set of 450 randomly generated test instances. These instances were generated after several preliminary computational experiments. In our proposed solution approach, we were able to find high-quality solutions within admissible computational times. In addition, the proposed matheuristic presents a low dependency on its parameters, pointing to the robustness of this developed solution procedure.

The computational experience pointed out that the proposed matheuristic outperformed the MILP model. In addition, the proposed RVFVS also outperformed SRSA metaheuristic – the most efficient solution procedure for the closest related variant available in the literature. The superiority of the proposed solution approach can be verified by means of the average relative percentage deviation and the success rate. In addition, we performed an ANOVA and a Tukey's range test to evaluate if this difference was statistically significant.

As suggestions for future research, the main research avenues can be highlighted. In the proposed solution approach, the decision variables were randomly selected to be fixed as zero in the sub-problem optimization. Thereby, we can consider fixation strategies that take problem characteristics into account. In addition, we can hybridize our variable neighbourhood search framework with other metaheuristics. For example, a tabu list can be considered to avoid the fixation of the same decision variables in two consecutive iterations of the search process. Another possibility is the consideration of an iterated greedy framework to destroy and build new fixation schemes into the incumbent solution. Recent studies have hybridized constraint programming and metaheuristics to solve production scheduling problems. The proposition of a constraint programming formulation and a hybrid algorithm is a promising research line.

Another suggestion for future studies is to consider other objective functions for the variant addressed in this paper, such as the total completion time or weighted earliness/tardiness penalties (just-in-time environment). The consideration of multi-objective variants of the production environment under study can be investigated. The explicit consideration of the periodical resource constraints in production scheduling problems is rather limited. We can also consider this type of constraint in other production environments, such as identical parallel machines, unrelated parallel machines, permutation flow shop, hybrid flow shop, job shop, or open shop.

ACKNOWLEDGMENT

This study was financed in part by the Coordination for the Improvement of Higher Education Personnel (CAPES), the São Paulo Research Foundation (FAPESP), through Grant no. #2021/11586-2 and the National Council for Scientific and Technological Development (CNPq), through grants 407151/2021-4 and 309755/2021-2.

REFERENCES

[1] M. Afzalirad and J. Rezaeian, "Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions," Computers & Industrial Engineering, vol. 98, pp. 40 – 52, 2016. https://doi.org/10.1016/j.cie.2016.05.020.

[2] M. Afzalirad and M. Shafipour, "Design of an efficient genetic algorithm for resource-constrained unrelated parallel machine scheduling problem with machine eligibility restrictions," Journal of Intelligent Manufacturing, vol. 29, no. 2, pp. 423–437, 2018. https://doi.org/10.1007/s10845-015-1117-6.

[3] M. Akbar and T. Irohara, "Scheduling for sustainable manufacturing: A review," Journal of cleaner production, vol. 205, pp. 866–883, 2018. https://doi.org/10.1016/j.jclepro.2018.09.100.

[4] M. P. Antonioli, C. D. Rodrigues, and B. A. Prata, "Minimizing total tardiness for the order scheduling problem with sequence-dependent setup times using hybrid matheuristics," International Journal of Industrial Engineering Computations, vol. 13, 2022. https://doi.org/10.5267/j.ijiec.2021.11.002.

[5] F. Della Croce and F. Salassa, "A variable neighborhood search based matheuristic for nurse rostering problems," Annals of Operations Re- search, vol. 218, no. 1, pp. 185–199, 2014. https://doi.org/10.1007/s10479-012-1235-x.

[6] F. Della Croce, V. Narayan, and R. Tadei, "The two-machine total completion time flow shop problem," European Journal of Operational Research, vol. 90, no. 2, pp. 227–237, 1996. https://doi.org/10.1016/0377-2217(95)00351-7.

[7] F. Della Croce, F. Salassa, and V. T'kindt, "A hybrid heuristic approach for single machine scheduling with release times," Computers & Operations Research, vol. 45, pp. 7–11, 2014. https://doi.org/10.1016/j.cor.2013.11.016.

[8] E. B. Edis and I. Ozkarahan, "Solution approaches for a real-life resource-constrained parallel machine scheduling problem," The International Journal of Advanced Manufacturing Technology, vol. 58, no. 9, pp. 1141–1153, 2012. https://doi.org/10.1007/s00170-011-3454-8.

[9] E. B. Edis, C. Oguz, and I. Ozkarahan, "Parallel machine scheduling with additional resources: Notation, classification, models and solution methods," European Journal of Operational Research, vol. 230, no. 3, pp. 449 – 463, 2013. https://doi.org/10.1016/j.ejor.2013.02.042.

[10] L. Fanjul-Peyro, F. Perea, and R. Ruiz, "Models and matheuristics for the unrelated parallel machine scheduling problem with additional resources," European Journal of Operational Research, vol. 260, no. 2, pp. 482 – 493, 2017. https://doi.org/10.1016/j.ejor.2017.01.002.

[11] V. Fernandez-Viagas and J. M. Framinan, "Neh-based heuristics for the permutation flowshop scheduling problem to minimise total tardiness," Computers & Operations Research, vol. 60, pp. 27–36, 2015. https://doi.org/10.1016/j.cor.2015.02.002.

[12] J. M. Framinan and P. Perez-Gonzalez, "Order scheduling with tardiness objective: Improved approximate solutions," European Journal of Operational Research, vol. 266, no. 3, pp. 840–850, 2018. https://doi.org/10.1016/j.ejor.2017.10.064.

[13] O. Herr and A. Goel, "Minimising total tardiness for a single machine scheduling problem with family setups and resource constraints," European Journal of Operational Research, vol. 248, no. 1, pp. 123–135, 2016. https://doi.org/10.1016/j.ejor.2015.07.001.

[14] M. Ji, J.-Y. Wang, and W.-C. Lee, "Minimizing resource consumption on uniform parallel machines with a bound on makespan," Computers & Operations Research, vol. 40, no. 12, pp. 2970 – 2974, 2013. https://doi.org/10.1016/j.cor.2013.06.011.

[15] S. Karhi and D. Shabtay, "Single machine scheduling to minimise resource consumption cost with a bound on scheduling plus due date assignment penalties," International Journal of Production Research, vol. 56, no. 9, pp. 3080–3096, 2018. https://doi.org/10.1080/00207543.2017.1400708.

[16] A. B. Keha, K. Khowala, and J. W. Fowler, "Mixed integer programming formulations for single machine scheduling problems," Computers & Industrial Engineering, vol. 56, no. 1, pp. 357–367, 2009. https://doi.org/10.1016/j.cie.2008.06.008.

[17] Y. Kochetov and A. Kondakov, "Vns matheuristic for a bin packing problem with a color constraint," Electronic Notes in Discrete Mathematics, vol. 58, pp. 39–46, 2017. https://doi.org/10.1016/j.endm.2017.03.006.

[18] S.-W. Lin and K.-C. Ying, "Optimization of makespan for no-wait flowshop scheduling problems using efficient matheuristics," Omega, vol. 64, pp. 115–125, 2016. https://doi.org/10.1016/j.omega.2015.12.002.

[19] M. Lubin and I. Dunning, "Computing in operations research using julia," INFORMS Journal on Computing, vol. 27, no. 2, pp. 238–248, 2015. https://doi.org/10.1287/ijoc.2014.0623.

[20] S. Martello, "Knapsack problems: algorithms and computer implementations," Wiley-Interscience series in discrete mathematics and optimization, 1990.

[21] N. Mladenovic and P. Hansen, "Variable neighborhood search," Computers & operations research, vol. 24, no. 11, pp. 1097–1100, 1997. https://doi.org/10.1016/S0305-0548(97)00031-2.

[22] J. C. Pinheiro and J. E. C. Arroyo, "Effective IG heuristics for a single-machine scheduling problem with family setups and resource constraints," Annals of Mathematics and Artificial Intelligence, vol. 88, no. 1-3, pp. 169–185, 2020. https://doi.org/10.1007/s10472-019-09646-6.

[23] A. R. Pitombeira-Neto and B. d. A. Prata, "A matheuristic algorithm for the one-dimensional cutting stock and scheduling problem with heterogeneous orders," TOP, Sep 2019. https://doi.org/10.1007/s11750-019-00531-3.

[24] B. A. Prata, A. R. Pitombeira-Neto, and C. J. M. Sales, "An integer linear programming model for the multiperiod production planning of precast concrete beams," Journal of Construction Engineering and Management, vol. 141, no. 10, p. 04015029, 2015. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000991.

[25] B. A. Prata, L. R. de Abreu, and J. Y. F. Lima, "Heuristic methods for the single-machine scheduling problem with periodical resource constraints," TOP, pp. 1–23, 2020. https://doi.org/10.1007/s11750-020-00574-x.

[26] B. A. Prata, C. D. Rodrigues, and J. M. Framinan, "Customer order scheduling problem to minimize makespan with sequence-dependent setup times," Computers & Industrial Engineering, vol. 151, p. 106962, 2021. https://doi.org/10.1016/j.cie.2020.106962.

[27] E. Rahimian, K. Akartunalı, and J. Levine, "A hybrid integer programming and variable neighbourhood search algorithm to solve nurse rostering problems," European Journal of Operational Research, vol. 258, no. 2, pp. 411–423, 2017. https://doi.org/10.1016/j.ejor.2016.09.030.

[28] G. Scheithauer, Introduction to Cutting and Packing Optimization: Problems, Modeling Approaches, Solution Methods. Springer, 2017, vol. 263. https://doi.org/10.1007/978-3-319-64403-5_1.

[29] D. Shabtay, "Single-machine scheduling with machine unavailability periods and resource dependent processing times," European Journal of Operational Research, 2021. https://doi.org/10.1016/j.ejor.2021.03.034.

[30] J. P. Sousa and L. A. Wolsey, "A time indexed formulation of non- preemptive single machine scheduling problems," Mathematical pro-gramming, vol. 54, no. 1-3, pp. 353–367, 1992. https://doi.org/10.1007/BF01586059.

[31] J. A. Ventura and D. Kim, "Parallel machine scheduling with earliness–tardiness penalties and additional resource constraints," Computers & Operations Research, vol. 30, no. 13, pp. 1945 – 1958, 2003. https://doi.org/10.1016/S0305-0548(02)00118-1.

[32] F. Villa, E. Vallada, and L. Fanjul-Peyro, "Heuristic algorithms for the unrelated parallel machine scheduling problem with one scarce additional resource," Expert Systems with Applications, vol. 93, pp. 28 – 38, 2018. https://doi.org/10.1016/j.eswa.2017.09.054.

[33] J.-B. Wang and M.-Z. Wang, "Single-machine scheduling to minimize total convex resource consumption with a constraint on total weighted flow time," Computers & Operations Research, vol. 39, no. 3, pp. 492 – 497, 2012. https://doi.org/10.1016/j.cor.2011.05.026.

[34] X.-R. Wang and J.-J. Wang, "Single-machine scheduling with convex resource dependent processing times and deteriorating jobs," Applied Mathematical Modelling, vol. 37, no. 4, pp. 2388 – 2393, 2013. https://doi.org/10.1016/j.apm.2012.05.025.

[35] L. Wu and C.-D. Cheng, "On single machine scheduling with resource constraint," Journal of Combinatorial Optimization, vol. 31, no. 2, pp. 491–505, 2016. https://doi.org/10.1007/s10878-014-9721-5.

[36] W.-C. Yeh, M.-C. Chuang, and W.-C. Lee, "Uniform parallel machine scheduling with resource consumption constraint," Applied Mathemati- cal Modelling, vol. 39, no. 8, pp. 2131 – 2138, 2015. https://doi.org/10.1016/j.apm.2014.10.012.

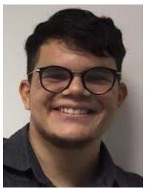
[37] Z. Zhu, F. Chu, L. Sun, and M. Liu, "Single machine scheduling with resource allocation and learning effect considering the rate-modifying activity," Applied Mathematical Modelling, vol. 37, no. 7, pp. 5371 – 5380, 2013. https://doi.org/10.1016/j.apm.2012.09.072.

NOMENCLATURE

i index for periods $\{1,2,...,m\}$.
j index for jobs $\{1,2,...,n\}$.
p_j processing time of job j.
r_j required resource of job j.
d_j due date of job j.
P maximum duration of the periods.
R maximum amount of resource for each period.
T_j tardiness of job j.
y_i if period i is used; 0, otherwise.
x_{ij} if job j is produced in period i; 0, otherwise.

AUTHOR'S BIOGRAPHY

Bruno de Athayde Prata has a Ph.D. in Industrial Engineering (University of Porto, Portugal). He is an Associate Professor in the Department of Industrial Engineering of Federal University of Ceara (Fortaleza, Brazil). His main research interests are combinatorial optimization problems, metaheuristics, and transportations systems. He regularly publishes papers in international scientific journals.



Levi Ribeiro de Abreu has a graduate in Industrial Engineering from the University of Ceará (UFC), Fortaleza, Ceara, Brazil, and is a Ph.D. student in Production Engineering from the University of Sao Paulo (USP), Sao Carlos, Sao Paulo, Brazil. His research is concentrated on Production Scheduling Problems with heuristic and mathematical programming approaches, with industrial applications in general.

Marcelo Seido Nagano is a Mechanical-Mechatronics Engineer, MSc, and Ph.D. in Production Engineering from the Sao Carlos School of Engineering, University of Sao Paulo, and a Postdoctoral Fellow in Mathematics and Applied Computing from the Associated Laboratory of Computing and Applied Mathematics of the National Institute for Space Research (INPE). He is currently a Productivity and Research Fellow at CNPq. He is a researcher in several areas of Operations Research applied to Production Systems, Knowledge Management, Innovation, and Entrepreneurship. He is a professor in the Undergraduate and Postgraduate programs in Production Engineering at the São Carlos School of Engineering, USP.