

Solving the 2-stage outsourcing flowshop problem through a MILP-heuristic

Roberto Tavares

Universidade Federal de São Carlos

Affiliation address

tavares@dep.ufscar.br

Fábio Molina da Silva

Universidade Federal de São Carlos

Affiliation address

fabio@dep.ufscar.br

RESUMO

A terceirização é uma estratégia comum usada por sistemas produtivos para aumentar sua capacidade sem investimentos adicionais no chão-de-fábrica. Normalmente, os tomadores de decisão se utilizam da terceirização para lidar com aumentos pontuais na demanda ou para aumentar métricas específicas. Esse estudo considera um ambiente flowshop de 2 estágios, onde operações do primeiro estágio podem ser terceirizadas. Esse problema é formulado através de um modelo de programação inteira mista e resolvido por uma abordagem baseada na técnica *branch-and-check*. Essa abordagem apresentou resultados interessantes quando aplicada em benchmarks existentes na literatura e se apresenta como uma estratégia viável para aplicações reais.

PALAVRAS CHAVE. Outsourcing, Scheduling, Branch and Check

Scheduling, Outsourcing, Heuristics

ABSTRACT

Outsourcing is a commonly used business tactic to increase production capacity without the need for additional investments in manufacturing facilities. Typically, decision makers use outsourcing to manage temporary spikes in demand or to improve specific performance metrics (such as delivery times) quickly and without making modifications to the manufacturing site. This study focuses on minimizing the total system makespan in a 2-stage flowshop setting where initial operation can be outsourced. The problem is formulated using Mixed Integer Linear Programming (MILP) and a heuristic based on the branch and check strategy is proposed. Although traditional MILP approaches are computationally intensive for various scheduling problems, the proposed branch and check approach presents interesting results in surpassing existing benchmarks in the literature and emerges as a practical strategy for real-world applications.

KEYWORDS. Scheduling. Outsourcing. Flowshop. Branch and check. Scheduling, Outsourcing, Heuristics

1. Introduction

Outsourcing is a business strategy that is frequently adopted by several industries. Recent studies focus on the adoption of this business strategy on remanufacturing and sustainability issues (for example, Wang et al. [2022b] presents carbon emission in a closed-loop supply chain considering remanufacturing and outsourcing), capacity planning [Haoues et al., 2021], electronics manufacturing, aircraft production, among others [Wang et al., 2022a]. Usually, the business strategy includes the outsourcing option to increase production capacity or improve a set of business metrics [Jiang et al., 2021; Lee and Sung, 2008a].

This paper focuses on operational issues of an outsourcing scenario. We consider a make-to-order manufacturing environment composed of a 2-machine flowshop, where the first processing operation can be either executed in-house or outsourced. If an operation is outsourced, there is a (financial) cost to be considered, and the job cannot be executed on the second stage before a supplier's lead time is elapsed. Thus, a solution is composed of three decisions: (i) if a job is outsourced (or not); (ii) the processing sequence of the nonoutsourced jobs on the first operation of the flowshop; (iii) the processing sequence on the second stage of the flowshop. The objective function is a weighted sum of the total outsource cost and the system's makespan.

Kim and Lee [2022] introduces this problem and states a set of important properties, and solve it by three different approaches (a Dynamic Programming algorithm, a Branch-and-Bound procedure (B&B) and a heuristic approach). The present research extends this work by proposing and implementing a mixed integer linear programming model (MILP) and a MILP-Heuristic approach. The heuristic approach could outperform the best exact method proposed by the literature by combining a critical path-based model on the master problem and an ordering algorithm on the slave procedure.

To present those methods, this research is structured as follows: Section 2 presents a background on scheduling problems with outsource allowed; Section 3 provides a formal description of the problem; section 4 contains the MILP formulation proposed for the problem; section 5 presents the hybrid MILP approach (H-MILP); both MILP and H-MILP were implemented and their results were compared with the B&B from Kim and Lee [2022]. The results are reported and analyzed in Section 6; final remarks are drawn in Section 7.

2. Outsourcing in Scheduling Problems

Outsourcing plays a crucial role in various business scenarios, offering decision-makers the opportunity to boost manufacturing capacity or improve specific manufacturing performance metrics (Lee et al., 2007, 2008; Jiang, 2021). Existing research in this field mainly focuses on either single or parallel machines. In the context of single-machine setups, Lee et al. (2007) introduced a set of dynamic programming and branch-and-bound algorithms designed to decrease the weighted total of flowtime and outsource cost on a single machine. They also defined an upper limit for the outsource cost, known as the "budget." Similarly, in single machine environments with a constrained outsourced budget, Lee et al. (2008) suggested a range of dynamic programming and branch-and-bound approaches to minimize both total lateness and tardiness. The study by Park (2017) also takes into account total completion time and total outsource cost as performance measures. Moreover, Wang et al. (2022) proposed MILP and Benders decomposition techniques to tackle the issue of minimizing total tardiness and outsource cost in a single machine, multi-period setting that allows preemption. Parallel machines are also addressed in the literature: Tavares Neto et al. (2015) present an Ant Colony Optimization method to minimize a weighted total of tardiness and outsource cost. Miao et al. (2017) examine job deterioration and outsourcing, with the aim of minimizing the weighted total of makespan and outsource costs. Chen et al. (2022) suggest leveraging outsourcing to enhance customer service levels and outsourcing costs by minimizing the total of production and outsourcing expenses.

In their work, Xu et al. [2021] examines the practice of outsourcing within a jobshop setting and introduces a Genetic Algorithm combined with a Tabu Search procedure. The goal is to reduce a weighted sum of cost, makespan, quality, and overall carbon emissions in the scheduling process. The topic of jobshop environments is also explored by Safarzadeh and Kianfar [2019], who suggest two precise methods (utilizing mathematical programming and constraint programming) to create schedules where tasks are completed internally or entirely outsourced.

In the literature, three categories of investigations are discussed concerning flowshop settings: the first one examines a flowshop where a job is either entirely outsourced or not (e.g., Choi and Chung [2011]; Tavares Neto and Godinho Filho [2011]; Chung and Choi [2013]). The second category permits outsourcing any operation of a job (e.g., Lee and Choi [2011]; Jiang et al. [2021]). The third category allows outsourcing only at one stage (e.g., Kim and Lee [2022]). The scholarly works employ two methods to determine the penalty for outsourcing a job: a fixed independent value (e.g., Kim and Lee [2022]; Tavares Neto and Godinho Filho [2011]) or a proportionate amount based on the in-house processing time (e.g., Lee and Choi [2011]; Jiang et al. [2021]). Additionally, some studies consider a significant transportation duration between the manufacturing facility and the subcontractors (e.g., refer to Qi [2011]).

When examining the strategies utilized for tackling scheduling challenges involving outsourcing, one may encounter methodologies that rely on constructive heuristics or metaheuristics (e.g., Xu et al. [2021]; Choi and Chung [2011]; Tavares Neto and Godinho Filho [2011]; Tavares Neto et al. [2015]), dynamic programming (e.g., Safarzadeh and Kianfar [2019]; Miao et al. [2017]; Lee and Sung [2008b,a]), MILP formulations (e.g., Wang et al. [2022a]; Tavares Neto and Godinho Filho [2011]; Tavares Neto et al. [2015]), Benders Decomposition [Wang et al., 2022a], and constraint programming [Safarzadeh and Kianfar, 2019]. In this review of the literature, we could not find any references of the adoption of a branch and check procedure being applied to scheduling problems involving outsourcing.

3. Problem definition

This research addresses the scenario outlined by Kim and Lee [2022]: a manufacturing setting consists of a 2-stage flowshop, where the initial stage processing can be done internally or externally. In case of outsourcing, the job is prepared for processing after a lead time l_i . Otherwise, the jobs are scheduled for internal processing and are prepared for the second stage based on their completion time. An outsourcing cost o_i is involved when a job is not handled in-house in the first stage. A weighting parameter δ is employed to harmonize the total outsourcing cost (OC) and the sequence completion time (CMAX) in the objective function (OF), as shown in Equation 1.

$$OF = \delta \cdot OC + (1 - \delta) \cdot CMAX \quad (1)$$

To define the problem, this paper assumes the notation presented in Tables 1 and 2. This study also adopts the following notation to address a job: if i is the job index, $[p]$ is the index of the job assigned to position p of a sequence.

The solving approaches presented in this study strongly incorporates the results of Johnson [1954]. In this classical algorithm, the authors presented a constructive heuristic to minimize makespan in a two-stage flowshop. One important property of this solution is that, given a Johnson's sequence $s_0, s_1, s_2, \dots, s_n$, if one element is removed, the remaining sequence is still a Johnson's sequence.

As presented by Kim and Lee [2022], this problem also has some important properties:

Property 1: A job j will not be outsourced if $\delta \cdot o_j \geq (1 - \delta) \cdot (f_j + q_j)$.

Symbol	Description
$J = \{0, \dots, J - 1\}$	Set of jobs
S_π	Set of jobs produced in-house (*)
O_π	Set of outsourced jobs (*)
$\{i, j\} \in J$	Index of a job
$P = \{0, \dots, J - 1\}$	Set of available positions on the in-house production sequence
$p \in P$	Index of a position
M	Very large number
f_i, q_i	Processing time of a job i on first and last stage, respectively
o_i	Outsource cost of job i
l_i	Leadtime of job i , if outsourced
$0 < \delta < 1$	A weight factor used to combine CMAX and OC on the objective function

(*) $J = S_\pi \cup O_\pi$

Tabela 1: Sets, index and constants used to describe a problem instance

Variable	Type	Description
OF	Continuous	Objective Function
OC	Continuous	Total outsourcing cost
CMAX	Continuous	System-wide makespan
x_{jp}	Binary	Indicates (has value 1) if job j is at position p of the production sequence
χ_{jp}	Binary	Indicates (has value 1) if the preprocessing stage allocates job j to be processed at position p
y_p	Binary	Indicates (has value 1) if job at position p is outsourced
y_j	Binary	Indicates (has value 1) if job j is outsourced
w_{ij}	Binary	Indicates (has value 1) if job j is processed after job i
S_j	Continuous	Moment when job j is ready to be processed on the second stage of the flowshop
C_p	Continuous	Completion time of job allocated at position p on the first stage of the flowshop
I_{kj}	Continuous	Idle time found on the stage $k \in \{0, 1\}$ before executing job j

Tabela 2: Variables used to describe a solution

Property 2: Given two jobs, i and j , it is more cost-beneficial to outsource j than i if $f_i < f_j$, $o_i \geq o_j$ and $l_i \geq l_j$.

Property 3: One of the optimal sequence of the in-house jobs respect Johnson's rule.

To solve this problem, this study presents two exact approaches: Section 4 formally defines the problem using a pure MILP formulation. However, when implemented in a state-of-the-art commercial solver, this approach was unable to solve problems with more than a few jobs. Therefore, Section 5 presents a MILP Heuristic based on the branch-and-check approach that could outperform the branch-and-bound algorithm presented in Kim and Lee [2022].

4. A Mixed-Integer Formulation

On this modeling approach, the sequence on the first machine of the jobs produced in-house is given by Johnson's rule [Kim and Lee, 2022]. Therefore, the decision at this stage is only whether to outsource or not the job at position p (variable y_p). The sequence obtained by Johnson's rule is represented by the matrix χ_{ip} (since this sequence does not change, it is a parameter of the model). Given the values of y_p , the completion times of the first stage are given by C_p . The sequence of jobs in the second processing stage is given by w_{ij} , and the processing start time is given by S_j . The model is composed by Equations 2-16.

$$z = \delta \cdot OC + (1 - \delta) \cdot CMAX \quad (2)$$

$$OC = \sum_p \left(\sum_j \chi_{jp} \cdot o_j \right) \cdot y_p \quad (3)$$

$$CMAX - S_j \geq q_j \quad \forall j \in J \quad (4)$$

$$\sum_{p=0}^{p'-1} \chi_{jp} \cdot C_p - \sum_{p=p'}^{|P|-1} \chi_{ip} \cdot C_p - M \cdot w_{ij} \leq 0 \quad \forall \begin{cases} p' \in P \\ j \in P \\ i \in P/j \end{cases} \quad (5)$$

$$C_p \geq y_p \cdot l_{[p]} \quad \forall p \in P \quad (6)$$

$$C_{p+1}^1 - C_p^1 \geq p_{[p+1]} \cdot (1 - y_{p+1}) \quad \forall p \in P/\{|P|\} \quad (7)$$

$$C_0^1 \geq p_{[0]} \cdot (1 - y_p) \quad (8)$$

$$C_p - C_p^1 + M \cdot y_p \geq 0 \quad \forall p \in P \quad (9)$$

$$S_j - \sum_{p \in P} \chi_{jp} \cdot C_p \geq 0 \quad j \in J \quad (10)$$

$$S_j - S_i - M \cdot (w_{ij} - 1) \geq q_i \quad \forall \begin{cases} j \in J/\{[0]\} \\ i \in J/\{j\} \end{cases} \quad (11)$$

$$w_{jj} = 0 \quad \forall j \in J \quad (12)$$

$$\sum_{i \in J} w_{ij} = 1 \quad \forall j \in J \quad (13)$$

$$\sum_{j \in J} w_{ij} = 1 \quad \forall i \in J \quad (14)$$

$$y_p = 0 \quad \forall p \in \{[i] \mid \delta \cdot o_i \geq (1 - \delta) \cdot (p_i + q_i)\} \quad (15)$$

$$y_p - y_{p'} \leq 0 \quad \forall \begin{cases} [p] \in J \\ [p'] \in J/\{[p]\} \\ p_{[p]} + q_{[p]} < p_{[p']} + q_{[p']} \\ o_{[p]} > o_{[p']} \\ l_{[p]} > l_{[p']} \end{cases} \quad (16)$$

On this formulation, the objective function is presented in Equation 2. The total outsource cost and makespan are found in Equations 3-4. The completion time of processing each job (at

each position p of the Johnson's sequence) is given by Equations 5-9. The starting times of the processing of the jobs on the second stage is found by Equations 10-11. Equation 12 establish that a job does not precede itself on the sequence of the second machine. Equations 15-16 are valid cuts derived from the study of Kim and Lee [2022]: Equations 15 represents Property 1, as reproduced in Section 3, and Equations 16 represents Property 2.

Although this formulation represents the problem and could achieve interesting results on small-sized instances, the amount of computational resources required to solve larger instances is prohibitive, specially when compared with the state-of-the-art exact methods found by the literature. Therefore, a LBBD approach is proposed in this study, as will be presented on Section 5.

5. A Branch and Check Approach

Several studies use this technique to address scheduling problems (e.g., see Hooker and Ottosson [2003]; Codato and Fischetti [2006]; Emde [2017]; Emde et al. [2020]). On this technique, the problem is decomposed into a Master Problem that is solved by a commercial MILP solver. On Branch and Check, the search space of the Master Problem contains all the solutions of the original problem. However, some of the solutions allowed by the Master Problem not necessarily exists on the original problem. Moreover, the objective function stated by the (original) Master problem also could need to be adjusted. Thus, for each solution found, a subproblem is solved and, if required, additional cuts are created on the Master problem. This procedure continues until the Master Problem finds the optimal solution and no new cuts are required.

In the Branch and Check approach proposed in this study, the Master Problem is presented in Section 5.1. A procedure to approach the subproblem is shown in Section 5.2.

5.1. The Master Problem

To formulate the Master Problem, this study chooses to adopt Equation 17 as a lower bound for C_{MAX} . This formulation is based on the work of Della Croce et al. [2000], and finds the makespan of the sequence ignoring any outsourced job by constraining C_{MAX} according to the critical path of the scheduling. Although it does not gives the final result of C_{MAX} (since the effects of the outsourced jobs on the second production stage are not considered), during the development of this solution, Equation 17 proves to be useful to guide the solver to a better solution.

$$\begin{aligned}
 C_{MAX} + \sum_{p'=0}^{p-1} \left(\sum_{i \in J} \chi_{ip'} \cdot p_i \right) \cdot y_{p'} + \left(\sum_{i \in J} \chi_{ip} \cdot p_i \right) \cdot y_p \\
 + \sum_{p'=p+1}^p \left(\sum_{i \in J} \chi_{ip'} \cdot p_i \right) \cdot y_{p'} \geq \sum_p \left(\sum_{i \in J} \chi_{ip} \cdot (p_i + q_i) \right) \quad \forall p \in P
 \end{aligned} \tag{17}$$

Therefore, the Master Model is composed by Equations 2, 3 and 17. The considerations regarding pre-ordering the jobs by Johnson's rules still apply.

5.2. The Subproblem

Once there is a new solution found by the Master Problem, the procedure to solve the subproblem is executed to adjust the value of C_{MAX} by using optimality cuts. To present this procedure, two sets are defined: a set of jobs produced in-house $S_\pi = \{j \in J | y_{[j]} = 0\}$, and the set of jobs outsourced $O_\pi = \{j \in J | y_{[j]} = 1\}$. Moreover, considering just the second production stage, it can be presented as a single operation where the jobs to be sequenced are available for production (or, following the scheduling literature, "released") on different times (either the outsource lead time or the completion time of the job on the previous stage). The objective is to minimize the makespan. This is the well-known $1/r_i/C_{MAX}$ problem, and can be solved by the Earliest Release Data (ERD) rule (i.e., sequence the jobs by r_i and program as soon as possible on the machine). Algorithm 1 presents the pseudo-code of the procedure used to solve the Slave Problem.

Algorithm 1 The pseudo-code for the Branch and Check approach

Vectors $q_i, r_i, i \in J$ Sets S_π and $O_\pi, S_\pi \cup O_\pi = J$ The makespan C_{MAX} found by the model
 Obtain the makespan C_{MAX}^* of this sequence by allocating each job on the second machine following the ERD rule

$C_{MAX} < C_{MAX}^*$ Create an optimality cut, as presented in Equations 18 - 19

This study proposes the optimality cuts presented in Equations 18 - 19. Both equations adjust the value of C_{MAX} of the Master Problem given the current values of the y_p variables.

$$C_{MAX} + \sum_{p \in S_\pi} (C_{MAX}^* + 1) \cdot y_p - \frac{C_{MAX}^*}{|O_\pi|} \cdot \sum_{p \in O_\pi} y_p \geq 0 \quad \text{If } |O_\pi| > 0 \quad (18)$$

$$C_{MAX} + \sum_{p \in S_\pi} C_{MAX}^* \cdot y_p \geq C_{MAX}^* \quad \text{If } |O_\pi| = 0 \quad (19)$$

6. Analysis and discussion

To analyze the viability of the methods proposed in this study, both exact methods (the model presented in Section 4 and the Branch and Check approach presented in Section 5) were implemented in Python 3, using CPLEX library. The Branch-and-Bound method proposed by Kim and Lee [2022] was implemented in C++.

A set of problem instances was generated following the description given by Kim and Lee [2022], and reported in Table 3 (on this table, $Uniform(a, b)$ states for a sample from a uniform distribution ranging from a to b). For each number of jobs ranging from $n = [10, 20, 30, 40, 50, 60]$, 20 problem instances were generated.

Parameter	Symbol	Range
Processing time (at each operation)	f_i, q_i	Uniform(1; 30)
Outsource cost	o_i	Uniform(1; 10)
Outsource leadtime	l_i	Uniform($0.1 \cdot \sum_{i \in N} f_i$; $0.3 \cdot \sum_{i \in N} q_i$)
Weight factor	δ	Uniform(0.3; 0.7)

Tabela 3: Parameters used to create a problem instance (based on Kim and Lee [2022])

The first analysis performed by this study was an aggregated analysis of the running time required to obtain the optimal solution by each method. This preliminary analysis shows that the pure MILP formulation was not able to outperform the B&B method proposed by Kim and Lee [2022] (not even for small-sized instances). Thus, this section focuses on the comparison between the B&B method and the proposed Branch and Check approach.

A subsequent analysis focuses on the performance charts presented in Figures 1 and 2. Figure 1 presents the performance chart of the B&B method. According to the data presented, the B&B method could obtain the optimal solution for most instances in less than 500s. However, there are problem instances that required almost 4,000s to be processed.

Figure 2 presents the results of both methods. The x-axis is limited on 182s, to enhance the differences between the B&B and the Branch and Check methods. The Branch and Check approach could solve all problems in 182s or less. Moreover, in less than 10s, the Branch and Check could obtain the optimal solution in most of the instances.

As shown in Figure 2, the Branch and Check approach outperform the B&B method on most instances. However, the plots showed that both methods were able to obtain the optimal solutions to some instances within a very short time. To better understand this behavior, Table 4 presents the results of the executions of both methods according to the instance size. Based on these

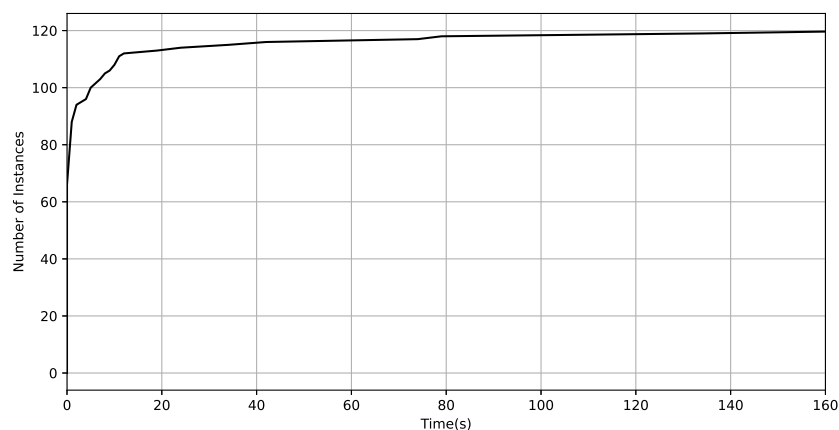


Figura 1: Performance chart of the Branch and Bound method

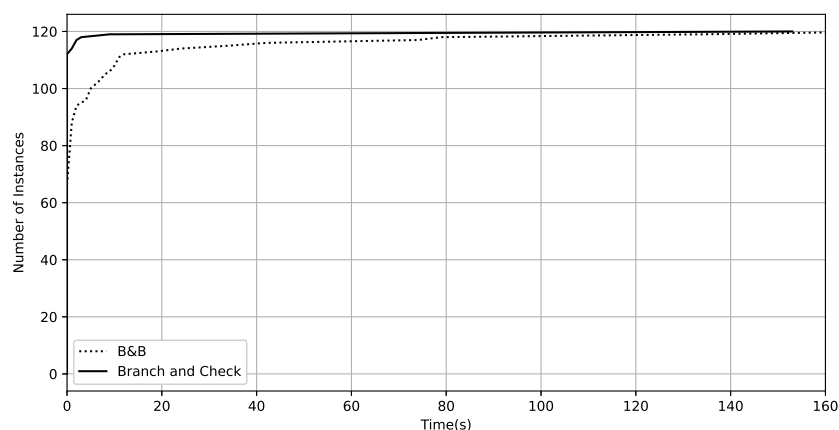


Figura 2: Performance chart of both methods

(0s-182s only)

data, it is possible to realize that, as expected, the B&B procedure requires more time accordingly to the number of jobs to be scheduled. A similar effect can be perceived when analyzing the number of cuts of the Branch and Check method: since each cut is created when an infeasible solution is found, the average number of invalid solutions found by CPLEX is not directly related only to the size of each instance.

7. Final remarks

This paper presents two exact approaches (a pure MILP formulation and a Branch and Check procedure) to solve the problem of minimize the weighted sum of makespan and total outsourcing cost on a 2-stage flowshop with possibility of outsource the first operation for some jobs. The first approach - a MILP formulation - could represent the problem properly, but could not outperform neither the Branch and Check or the B&B algorithm presented in the literature.

The Branch and Check procedure proposed on this paper considers a Master problem formulation based on assignment variables and a critical path modeling of the not outsourced jobs.

Size	B&B		LBBD	
	Mean Time (s)	Number Of Nodes (avg.)	Mean Time (s)	Number of cuts (avg.)
10	0.00	498	0.06	52.50
20	0.10	17,683	0.47	160.65
30	0.50	272,990	0.20	65.55
40	0.90	686,612	0.42	98.20
50	14.35	9,208,852	3.13	312.45
60	21.75	11,638,193	0.31	37.50

Tabela 4: Results of the computational experiments

To avoid the need of sequencing the not outsourced jobs, it is proposed to apply the Johnson's rule previously. The subproblem is then solved to perform the fine adjustment on the objective function value.

To assess the behavior of the proposed Branch and Check procedure, we implemented the best exact method found for this problem (a B&B algorithm). Although both methods result in a similar behavior for instances with 40 or less jobs, the Branch and Check outperform the B&B algorithm when applied on larger instances (e.g., the mean time required by the B&B to solve a 90-jobs instance is 1961,65s, and the Branch and Check required a mean time of 3.77s).

We believe that the findings of this research can be used for inspiration on real scenarios with similar modeling: although exact methods are usually outperformed by the speed of (meta-)heuristics approach - specially in short-term planning such as shop floor scheduling problems - our Branch and Check approach could obtain optimal solutions in a very reasonable time stamp.

Acknowledgments

We would like to express our gratitude to the São Paulo Research Foundation (FAPESP) for their financial support through Grant No. 2016/01860-1.

Referências

- Choi, B.-C. and Chung, J. (2011). Two-machine flow shop scheduling problem with an outsourcing option. *European Journal of Operational Research*, 213(1):66 – 72. ISSN 03772217. URL <http://dx.doi.org/10.1016/j.ejor.2011.03.017>.
- Chung, D.-Y. and Choi, B.-C. (2013). Outsourcing and scheduling for two-machine ordered flow shop scheduling problems. *European Journal of Operational Research*, 226(1):46 – 52. ISSN 03772217. URL <http://dx.doi.org/10.1016/j.ejor.2012.10.048>.
- Codato, G. and Fischetti, M. (2006). Combinatorial benders' cuts for mixed-integer linear programming. *Operations Research*, 54(4):756–766.
- Della Croce, F., Gupta, J. N., and Tadei, R. (2000). Minimizing tardy jobs in a flowshop with common due date. *European Journal of Operational Research*, 120(2):375–381. ISSN 0377-2217. URL <https://www.sciencedirect.com/science/article/pii/S0377221799001642>.
- Emde, S. (2017). Optimally scheduling interfering and non-interfering cranes. *Naval Research Logistics (NRL)*, 64(6):476–489. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.21768>.
- Emde, S., Polten, L., and Gendreau, M. (2020). Logic-based benders decomposition for scheduling a batching machine. *Computers & Operations Research*, 113:104777. ISSN 0305-0548.

- Haoues, M., Dahane, M., and Mouss, K. N. (2021). Capacity planning with outsourcing opportunities under reliability and maintenance constraints. *International Journal of Industrial and Systems Engineering*, 37(3):382–409.
- Hooker, J. N. and Ottosson, G. (2003). Logic-based benders decomposition. *Mathematical Programming*, 96(1):33–60. ISSN 1436-4646. URL <https://doi.org/10.1007/s10107-003-0375-9>.
- Jiang, X., Zhang, A., Chen, Y., Chen, G., and Lee, K. (2021). An improved algorithm for a two-stage production scheduling problem with an outsourcing option. *Theoretical Computer Science*, 876: 59 – 69. ISSN 03043975. URL <http://dx.doi.org/10.1016/j.tcs.2021.05.022>.
- Johnson, S. M. (1954). Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, 1(1):61–68. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800010110>.
- Kim, E.-S. and Lee, I. S. (2022). Scheduling of two-machine flowshop with outsourcing lead-time. *Computers & Operations Research*, 145:105864. ISSN 0305-0548. URL <https://www.sciencedirect.com/science/article/pii/S0305054822001381>.
- Lee, I. S. and Sung, C. (2008a). Minimizing due date related measures for a single machine scheduling problem with outsourcing allowed. *European Journal of Operational Research*, 186(3):931 – 952. ISSN 03772217. URL <http://dx.doi.org/10.1016/j.ejor.2007.02.015>.
- Lee, I. S. and Sung, C. (2008b). Single machine scheduling with outsourcing allowed. *International Journal of Production Economics*, 111(2):623 – 634. ISSN 09255273. URL <http://dx.doi.org/10.1016/j.ijpe.2007.02.036>.
- Lee, K. and Choi, B.-C. (2011). Two-stage production scheduling with an outsourcing option. *European Journal of Operational Research*, 213(3):489 – 497. URL <http://dx.doi.org/10.1016/j.ejor.2011.03.037>.
- Miao, C., Meng, F., Zou, J., and Jia, B. (2017). Batch scheduling with proportional-linear deterioration and outsourcing. *Mathematical Problems in Engineering*, 2017. ISSN 1024123X. URL <http://dx.doi.org/10.1155/2017/6093029>.
- Qi, X. (2011). Outsourcing and production scheduling for a two-stage flow shop. *International Journal of Production Economics*, 129(1):43 – 50. URL <http://dx.doi.org/10.1016/j.ijpe.2010.08.011>.
- Safarzadeh, H. and Kianfar, F. (2019). Job shop scheduling with the option of jobs outsourcing. *International Journal of Production Research*, 57(10):3255 – 3272. ISSN 00207543. URL <http://dx.doi.org/10.1080/00207543.2019.1579934>.
- Tavares Neto, R. and Godinho Filho, M. (2011). An ant colony optimization approach to a permutational flowshop scheduling problem with outsourcing allowed. *Computers and Operations Research*, 38(9):1286 – 1293. URL <http://dx.doi.org/10.1016/j.cor.2010.11.010>.
- Tavares Neto, R. F., Godinho Filho, M., and da Silva, F. M. (2015). An ant colony optimization approach for the parallel machine scheduling problem with outsourcing allowed. *Journal of Intelligent Manufacturing*, 26(3):527 – 538. ISSN 09565515. URL <http://dx.doi.org/10.1007/s10845-013-0811-5>.

- Wang, L., Zhang, Z., and Yin, Y. (2022a). Order acceptance and scheduling problem with outsourcing in seru production system considering lot-spitting. *European Journal of Industrial Engineering*, 16(1):91 – 116. ISSN 17515254. URL <http://dx.doi.org/10.1504/EJIE.2022.119371>.
- Wang, N., Zhang, Y., and Li, J. (2022b). Carbon emission reduction and coordination in a closed-loop supply chain with outsourcing remanufacturing. *Kybernetes*, 51(11):3366 – 3393. ISSN 0368492X. URL <http://dx.doi.org/10.1108/K-11-2020-0800>.
- Xu, W., Hu, Y., Luo, W., Wang, L., and Wu, R. (2021). A multi-objective scheduling method for distributed and flexible job shop based on hybrid genetic algorithm and tabu search considering operation outsourcing and carbon emission. *Computers and Industrial Engineering*, 157. ISSN 03608352. URL <http://dx.doi.org/10.1016/j.cie.2021.107318>.