A hybrid metaheuristic with a splitting-based procedure for the Parallel Batch Processing Machine Scheduling problem

Maria Alejandra Moncada [1] and Abelino Sepúlveda[2]

Advisor:
Juan Carlos Rivera[3]

Research practice II
Research proposal
Mathematical Engineering
School of Applied Sciences and Engineering.
Universidad EAFIT

July 2022

[1]Mathematical Engineering student. Universidad EAFIT, Colombia. mamoncadaa@eafit.edu.co.
[2]Mathematical Engineering student. Universidad EAFIT, Colombia. asepulvede@eafit.edu.co.
[3]Mathematical Modeling Research Group. Mathematical Science Department. School of Applied Sciences and Engineering. Universidad EAFIT. jrivera6@eafit.edu.co

1 Introduction

The Batch scheduling problem, also known as Batch Processing Machine (BPM), is a typical combinatorial optimization problem. In BPM, A batch is a group of jobs that a machine can process simultaneously and then release together. This problem is encountered in many industries such as the manufacturing industry (Fanti *et al.*, 1996b), furniture manufacturing industry (Yuan *et al.*, 2004), metal industry (Ram & Patel, 1998), casting industry (Mathirajan *et al.*, 2004), and aircraft industry (Van De Rzee *et al.*, 1997). Therefore, solving this problem has significant economic impacts.

For the BPM problem, there are two variants. First, the single batch processing machine proposed by Ikura & Gimple (1986), who studied the problem with constant batch processing time and identical job sizes, has approached with different processing times and release dates as proposed by Cheraghi *et al.* (2003). The second is the parallel batch processing machine introduced by Lee *et al.* (1992), who suggested that two or more identical machines with the same capacity can process the batches. For our purpose, we employ the parallel BPM with non-identical size Jobs. However, Uzsoy (1994) proves that the scheduling problem minimizing the makespan on a single BPM with non-identical job sizes is NP-hard, and grouping jobs into batches is equivalent to the bin-packing problem that is NP-hard (Garey & Johnson, 1979). Therefore, the parallel BPM is also an NP-hard problem.

In recent years, the search for methodologies to solve the parallel BPM has grown significantly due to its potential application in the industry. Nevertheless, solution techniques have not been explored in this kind of application. For instance, Cortés Zapata *et al.* (2018) proposed a splitting algorithm to translate a sequence of jobs to a BPM solution by grouping jobs into batches. We intend to adapt this type of solution strategy to a more complicated scheduling problem with parallel machines, which implies at least two decision levels as explained by Rivera *et al.* (2013) for a multi-trip cumulative capacitated vehicle routing problem.

2 Statement of the problem

2.1 Statement of the problem

Manufacturing is the creation or production of goods with the help of equipment, labor, machines, tools, and chemical or biological processing or formulation. Manufacturing optimization is a broad discipline that enables producers to move as rapidly and waste-free as feasible, from prototyping through mass production. It is a data-driven acceptance of a superior method that uses cutting-edge technology and mathematics robust foundation to support it. Manufacturing businesses nowadays face a variety of difficulties. The reduction of production costs and increased timeliness of deliveries to their customers can set the course for long-term competitiveness in the market. As these processes frequently operate as the bottleneck in the manufacturing process, scheduling batch processing equipment will significantly improve task completion times.

Numerous industrial applications and complex solutions to real-world problems have driven much research on scheduling issues in batch processing equipment. Batch processing machine scheduling is a common combinatorial optimization problem. In contrast to conventional scheduling issues, a batch processing system may handle multiple workloads simultaneously. In order to make the duties easier and reduce the time spent handling the material, the key objective in these difficulties is to batch jobs together and process them simultaneously in a machine.

The problem stated consists of programming a set $J = \{1, \ldots, n\}$ of n jobs in a BPM with a maximum capacity of B. Every job $j \in J$ has a release time r_j, a minimum processing time p_j, and a size s_j. Jobs can be grouped in batches where the sum of the sizes s_j does not overcome the capacity of machine B. The objective is to minimize the necessary time to process the n jobs (C_{max}). The operations research (OR) literature describes two types of batching: serial batching (s-batch) and parallel batching (p-batch). On the other hand, the BPM problem can be divided into four main categories, using the Graham *et al.* (1979) notation:

- $1|batch, B|C_{max}$: Does not consider different release times or different sizes of jobs.
- $1|batch, r_j, B|$: Consider different release times, but no different sizes of jobs.
- $1|batch, \sum_j s_j \leq B|$: Consider different sizes of jobs, but no different release times.
- $1|batch, r_j, \sum_j s_j \leq B|$: Consider both different release times and different sizes of jobs.

This project will focus on parallel batching and jobs with arbitrary released time and size. In parallel batching, multiple jobs are processed simultaneously on a batch processing machine (BPM). $1|batch, r_j, \sum_j s_j \leq B|$ is an NP-hard problem. Because of this, the use of different heuristic methods is expected. Many heuristics were put forth, including First Fit Longest Processing Time (FFLPT) and First Fit Shortest Processing Time (FFSPT). Two efficient heuristics—best fit longest processing time (BFLPT) and sequential knapsack problem (SKP)—were first presented by Dupont & Ghazvini (1998), and subsequently, one exceeded FFLPT. Dupont & Dhaenens-Flipo (2002), who presented some dominance properties for a universal enumeration scheme for the makespan criteria, offered an exact algorithm like branch and bind to solve the problem optimally. An enumeration technique was created and integrated by Li *et al.* (2016) with already existing heuristics to address complex problems (Li & Wang, 2018).

2.2 Formalization of the problem

BPM can be mathematically formulated as a Mixed-integer linear programming (MILP) model as follows (Trindade *et al.*, 2018).
Given a set $J := \{1, \ldots, n_J\}$ of jobs, each job $j \in J$ has a processing time p_j and a size s_j. Each of them must be assigned to a batch $k \in K := \{1, \ldots, n_K\}$, the sum of the sizes of the jobs assigned to a single batch cannot exceed B. It is assumed that $s_j \leq B$, for all $j \in J$.It is also assume that the batches should be assigned to a specific machine $m \in M := \{1, \ldots, n_M\}$. The processing time P_k of each batch $k \in K$ is defined as longest processing time among all jobs assigned to it, i.e., $P_k := \max\{p_j : j \text{ is assigned to } k\}$. Binary variables take the value of one when the job $j \in J$ is processed in the batch $k \in K$ and machine $m \in M$ and zero otherwise:

$$x_{jkm} = \begin{cases} 1, & \text{if job } j \text{ is assigned to batch } k \text{ in the machine } m. \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Also for every batch $k \in K$ a P_{km} and S_{km} variables are defined:

$$P_{km}: \text{ time to process batch } k \text{ in machine } m. \tag{2}$$

$$S_{km}: \text{ time when batch } k\text{starts to be processed in machine } m. \tag{3}$$

Mentioned above was used by Trindade *et al.* (2018) to formulate the following equations:

$$\min\ C_{\max} = S_{nk} + P_{nk} \tag{4}$$

$$\tag{5}$$

$$\text{s.a.} \sum_{j\in J} s_j \cdot x_{jkm} \le B \qquad \forall\ k \in K,\ m \in M \tag{6}$$

$$\sum_{k\in K}\sum_{m\in M} x_{jkm} = 1, \qquad \forall\ j \in J \tag{7}$$

$$\tag{8}$$

$$C_{max} \ge 0, \tag{9}$$

$$x_{jkm} \in \{0,1\}, \qquad \forall j \in J, \forall k \in K \tag{10}$$

$$S_{km} \ge r_j x_{jkm}, \qquad \forall j \in J, \forall k \in K, \forall m \in M \tag{11}$$

$$S_{km} \ge S_{(k-1)m} + P_{(k-1)m}, \qquad \forall j \in J, \forall k \in K : k > 1, \forall m \in M \tag{12}$$

$$S_{km} \ge 0, \qquad \forall k \in K, \forall m \in M \tag{13}$$

$$C_{max} \ge S_{km} + p_j x_{nmk}, \qquad \forall m \in M \tag{14}$$

The equation 4 refers to the objective function that minimizes the makespan. Constraints 6 ensure that each batch respects the machine's capacity. The equation 7 is a constraint that ensures that each job is assigned to a single batch and a single machine. Equation 9 denotes that the makespan must be more than zero. Constraints 10 determine whether a job can only be assigned or not. Constraints 11-13 determine when each machine m starts to process each batch k. Constraints 14 determine the makespan(Trindade *et al.*, 2018).

3 Objectives

3.1 General objective

Develop a hybrid metaheuristic with a splitting-based procedure for the parallel batch processing machine scheduling problem.

3.2 Specific objectives

- Review the state of the art about splitting algorithm and its applications.
- Review the state of the art about solution approaches for the Parallel Batch Processing Machine and related problems.
- Develop a splitting algorithm for the Parallel Batch Processing Machine.
- Develop a hybrid metaheuristic for the Parallel Batch Processing Machine.
- Validate and compare solutions with benchmark results in order to evaluate the proposed algorithm performance.
- Write a research paper to socialize the developed models.

4 Justification

BPM is an NP-hard problem. A problem H is NP-hard when every problem L in NP can be reduced in polynomial time to H; assuming a solution for H takes 1 unit time, H's solution can be used to solve L in polynomial time. This means that an optimal solution is unobtainable. If the optimal solution is unattainable, then it is reasonable to sacrifice optimality and settle for a "good" feasible solution that can be computed efficiently. Of course, we would like to sacrifice as little optimality as possible while gaining as much as possible in efficiency (Hochba, 1997). Because of that is important to keep searching for heuristic methods to find better solutions.

BPM has multiple applications in diverse industries. These include manufacturing (Fanti *et al.*, 1996b), furniture manufacturing (Yuan *et al.*, 2004), shoe manufacturing (Fanti *et al.*, 1996a), metal industry (Ram & Patel, 1998), casting industry (Mathirajan *et al.*, 2004),aircraft industry (Van De Rzee *et al.*, 1997), automobile gear manufacturing (Gokhale & Mathirajan, 2011),and healthcare(Ozturk *et al.*, 2011).This problem can also be found in chemical processes in tanks or kilns. Nevertheless, the bulk of the literature on p-batch scheduling deals with the semiconductor industry (Fowler & Mönch, 2022). Since this problem has many applications, finding a better solution than the already proposed can improve the economy of many industries.

Multiple heuristic methods have been applied to BPM. As mention before for the solution to this problem, Li *et al.* (2005) developed an approximation algorithm $(2+e)$. Contrastingly Melouk *et al.* (2004) worked with Simulated annealing (SA)and the software CPLEX. On the other hand, Chou *et al.* (2006) divided the problem into two phases. In the first one, Genetic Algorithm (GA) was used, and First-Fit Longest Processing Time (FFLPT) was used in the second one. Velez-Gallego *et al.* (2011) used an extension of the SKP(Successive Knapsack Problem heuristic as a constructive solution of the $1|batch, \sum_j s_j \leq B|$. Additionally, in the literature, many variables of the BPM can be found with different solution methods. However, the adaptation of this problem to the shortest path problem using the Two-Level Split algorithm has not been made before. This method can bring better solutions than the already found ones since this algorithm is very effective for routing problems.

5 Scope

For the development of the heuristic algorithms proposed to solve the problem, we have access to public data that will facilitate the development, validation, and verification of the models. Additionally, There is access to free libraries in the programming language where we will develop the algorithm to solve the problem. On the other hand, the results will be compared with the data published in the literature. It gives us many limitations to measuring the algorithm's efficiency compared to others developed since we do not have the source code of these, and the implementations will not be carried out either.

The development of efficient algorithms for parallel BPM can significantly help, thanks to its diverse applicability in the industry. For this reason, one of the main expected results of this research is that the algorithm developed for parallel BPM obtains feasible competitive solutions compared to the algorithms available in the literature.

6 State of the art

Due to the NP-hard nature of the BPM problem, no precise algorithms can guarantee the best results in polynomial time (Chandru *et al.*, 1993). In numerous approaches, efficient metaheuristic algorithms are proposed and implemented to guarantee solutions close to the optimum. Nevertheless, non-identical job sizes or release times are rarely considered in research. We review studies that primarily focused on that because of this.

Trindade *et al.* (2020) proposed an arc-flow-based model for minimizing makespan on a parallel processing machine. On their arc-flow formulation, consider jobs with non-identical sizes for parallel machines turning it into an NP-hard problem. The idea is to formulate a model using a graph compression technique. In the graph, each physical space of the batch with is represented by a node. Once they propose the model solving the problem computationally, showing that the results of the proposed model have a better performance than others found in the literature.

An investigation by Xu *et al.* (2012) proposes a heuristic model and an ant colony heuristic algorithm (ACO). They develop a mixed integer linear programming model and propound a lower bound to evaluate the performance of the proposed algorithms. Furthermore, they introduced in the model a strategy to achieve a satisfactory solution in a reasonable computational time. The strategy consists of introducing a candidate list that restricts the number of possible eligible candidates to be considered in each construction step. The results show that the implementation of the ACO with its candidate list strategy was more robust and consistent than the other heuristics proposed.

The development of heuristic algorithms has had a large field of research and development over the years. However, it is well known that Genetic Algorithms (GA) have been among the most popular for solving problems, especially NP-hard ones. Since it was proposed in the '70s by Holland (1975), the development of these algorithms has become famous. That is why the BPM also has approached this algorithm. One of those approaches was presented by Balasubramanian *et al.* (2004), who developed two different versions of the Genetic Algorithm. The results show that the results obtained by the algorithms of the first version are better than the other developed version. The first version consists of forming the batches and then deciding which machines to assign them. Another investigation developed by Arroyo & Leung (2017) introduced GA and several heuristics based on the first-fit and best-fit job setup time rules and a lower bound to evaluate the algorithm's quality. Finally, the results obtained are analyzed using the Relative Percentage Deviation (RPD) between the heuristic solutions and the solutions of the mixed integer linear optimization problem statement.

Finally, we were in search of applications and methodologies for splitting-based algorithms. Since this algorithm was first proposed by Prins (2004a), it has been applied to the VRP family of vehicle routing problems. The first and unique implementation of a splitting-based heuristic algorithm for batch machine processing is proposed by Cortés Zapata *et al.* (2018). They convert the single BPM into the shortest path problem to find an optimal solution to the problem.

7 Proposed methodology

For the BPM problem, Li *et al.* (2005) presents an approximation algorithm with a worst-case ratio of $2 + \epsilon$, where $\epsilon > 0$ can be arbitrarily small. Contrastingly Melouk *et al.* (2004) worked simulated annealing (SA) approach to minimize makespan for a single batch-processing machine. Each job has a corresponding processing time and size. They compare the results of the SA approach to CPLEX software. On the other hand, Chou *et al.* (2006) presents two versions of a hybrid genetic algorithm (GA). They proposed an improved technique (merge-split procedure) to refine the answer provided by the LPT-BFF heuristic. Li & Zhang (2018) present a mixed integer programming (MIP) model for this problem. To address this issue, they also create heuristic algorithms, such as the biased random-key genetic algorithm (BRKGA) and the hybrid bin loading (HBL) algorithm. An extension of the SKP (Successive Knapsack Problem) heuristic, as a constructive solution of the $1|batch, \sum_j s_j \leq B|$, is used by Velez-Gallego *et al.* (2011).

The Split algorithm was proposed for the first time by Prins (2004b) for vehicle routing problems (VRP). This algorithm is also seen in the literature as order-first cluster-second. In its traditional version, given a sequence of nodes, called a "giant tour", the optimal solution is obtained by inserting visits to the deposit. The Split has been used before in problem as Capacitated Vehicle Routing Problem (CVRP)(Beasley, 1983), Multi-compartment VRP (Fallahi *et al.*, 2008), Capacitated arc routing problem(CARP) (Lacomme *et al.*, 2004), multistart iterated local search for the multitrip cumulative capacitated vehicle routing problem (Rivera *et al.*, 2015) and the Truck and Trailer Routing Problem(TTRP) (Villegas *et al.*, 2013). Rivera *et al.* (2013) proposed a variation called Two-Level Split that, in routing problems, works by reformulating the problem to a set of shortest path problems. Since the BPM can be formulated as a shortest path problem, the Two-Level Split can be adapted to the split algorithm.

The main objective of developing a splitting-based algorithm to the BPM problem is because it has been applied to this family of problems and because it has the benefit of being easily implemented with other heuristic and metaheuristic algorithms. This enables the creation of more effective algorithms to address the issue. Duhamel *et al.* (2012) presented a hybrid Evolutionary Local Search (hybrid ELS) for non-homogeneous fleet Vehicle Routing Problems, which included some of these implementations. As the first genetic algorithm capable of competing with the most successful strategies at the time, tabu search heuristics, Prins (2004b) proposed a similar strategy for the CVRP. Finally, the iterative multi-start evolutionary local search (MS-ELS) algorithm proposed by Villegas *et al.* (2010) is another intriguing application of the split algorithm.

Finally, once the split algorithm has been implemented, it is planned to implement algorithms such as those mentioned above together with split. Such as the multi-start evolutionary local search Villegas *et al.* (2010), ant colony optimization algorithm Santos *et al.* (2010), large neighborhood Tang *et al.* (2009) or the implementation of genetic algorithms such as NSGA-II. The implementation of these algorithms is beneficial, since they allow the development of efficient solutions to the BPM problem through local or neighborhood searches, thus avoiding local minima and being able to find global solutions for the problem.

8 Schedule, commitments and deliverables

To the develop of the methodology describe in the section 7, weekly meetings will be done.

Table 1 summarizes the proposed schedule for this research practice.

Table 1: Schedule

Activity	*Weeks*																	
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*
Conceptual model	■	■	■															
Mathematical model and solution algorithms				■	■	■	■	■	■	■	■	■						
Validation and verification of results													■	■	■			
Formalize paper																■	■	■

9 Intellectual property

According to the internal regulation on intellectual property within Universidad EAFIT, the results of this research practice are product of *Maria Alejandra Moncada*, *Abelino Sepúlveda* and *Juan Carlos Rivera*.

In case further products, beside academic articles, that could be generated from this work, the intellectual property distribution related to them will be directed under the current regulation of this matter determined by Universidad EAFIT (2017).

References

Arroyo, José Elias C, & Leung, Joseph Y-T. 2017. Scheduling unrelated parallel batch processing machines with non-identical job sizes and unequal ready times. *Computers & Operations Research*, **78**, 117–128.

Balasubramanian, Hari, Mönch, Lars, Fowler, John, & Pfund, Michele. 2004. Genetic algorithm based scheduling of parallel batch machines with incompatible job families to minimize total weighted tardiness. *International Journal of Production Research*, **42**(8), 1621–1638.

Beasley, JE. 1983. Route first–Cluster second methods for vehicle routing. *Omega*, **11**(4), 403–408.

Chandru, Vijaya, Lee, Chung-Yee, & Uzsoy, Reha. 1993. Minimizing total completion time on a batch processing machine with job families. *Operations Research Letters*, **13**(2), 61–65.

Cheraghi, SH, Vishwaram, Vishwaram, & Krishnan, Krishna K. 2003. Scheduling a single batch-processing machine with disagreeable ready times and due dates. *International Journal of Industrial Engineering: Theory, Applications and Practice*, **10**(2), 175–187.

Chou, Fuh-Der, Chang, Pei-Chann, & Wang, Hui-Mei. 2006. A hybrid genetic algorithm to minimize makespan for the single batch machine dynamic scheduling problem. *International Journal of Advanced Manufacturing Technology*, **31**(01), 350–359.

Cortés Zapata, Ana María, *et al.* 2018. *Algoritmos metaheurísticos híbridos para el problema de máquinas de procesamiento por lotes*. Ph.D. thesis, Universidad EAFIT.

Duhamel, Christophe, Lacomme, Philippe, & Prodhon, Caroline. 2012. A hybrid evolutionary local search with depth first search split procedure for the heterogeneous vehicle routing problems.

Engineering Applications of Artificial Intelligence, **25**(2), 345–358. Special Section: Local Search Algorithms for Real-World Scheduling and Planning.

Dupont, Lionel, & Dhaenens-Flipo, Clarisse. 2002. Minimizing the makespan on a batch machine with non-identical job sizes: an exact procedure. *Computers Operations Research*, **29**(7), 807–819.

Dupont, Lionel, & Ghazvini, Fariborz Jolai. 1998. Minimizing makespan on a single batch processing machine with non-identical job sizes. *Journal européen des systèmes automatisés*, **32**(4), 431–440.

Fallahi, Abdellah El, Prins, Christian, & Wolfler Calvo, Roberto. 2008. A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. *Computers Operations Research*, **35**(5), 1725–1741. Part Special Issue: Algorithms and Computational Methods in Feasibility and Infeasibility.

Fanti, M. P., Maione, B., Piscitelli, G., & Turchiano, B. 1996a. Heuristic scheduling of jobs on a multi-product batch processing machine. *International Journal of Production Research*, **34**(8), 2163–2186.

Fanti, MP, Maione, B, Piscitelli, G, & Turchiano, B. 1996b. Heuristic scheduling of jobs on a multi-product batch processing machine. *International Journal of Production Research*, **34**(8), 2163–2186.

Fowler, John W., & Mönch, Lars. 2022. A survey of scheduling with parallel batch (p-batch) processing. *European Journal of Operational Research*, **298**(1), 1–24.

Garey, Michael R, & Johnson, David S. 1979. Computers and intractability. *A Guide to the Theory of NP-Completeness.*

Gokhale, Ravindra, & Mathirajan, M. 2011. Heuristic algorithms for scheduling of a batch processor in automobile gear manufacturing. *International Journal of Production Research*, **49**(10), 2705–2728.

Graham, R.L., Lawler, E.L., Lenstra, J.K., & Kan, A.H.G.Rinnooy. 1979. Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey. *Pages 287–326 of:* Hammer, P.L., Johnson, E.L., & Korte, B.H. (eds), *Discrete Optimization II.* Annals of Discrete Mathematics, vol. 5. Elsevier.

Hochba, Dorit S. 1997. Approximation algorithms for NP-hard problems. *ACM Sigact News*, **28**(2), 40–52.

Holland, John Henry. 1975. *Adaption in natural and adaptive systems.*

Ikura, Yoshiro, & Gimple, Mark. 1986. Efficient scheduling algorithms for a single batch processing machine. *Operations Research Letters*, **5**(2), 61–65.

Lacomme, Philippe, Prins, Christian, & Ramdane Cherif-Khettaf, Wahiba. 2004. Competitive Memetic Algorithms for Arc Routing Problems. *Annals OR*, **131**(10), 159–185.

Lee, Chung-Yee, Uzsoy, Reha, & Martin-Vega, Louis A. 1992. Efficient algorithms for scheduling semiconductor burn-in operations. *Operations Research*, **40**(4), 764–775.

Li, Shuguang, Li, Guojun, Wang, Xiaoli, & Liu, Qiming. 2005. Minimizing makespan on a single batching machine with release times and non-identical job sizes. *Operations Research Letters*, **33**(2), 157–164.

Li, XiaoLin, & Wang, Yu. 2018. Scheduling Batch Processing Machine Using Max–Min Ant System Algorithm Improved by a Local Search Method. *Mathematical Problems in Engineering*, **2018**(01), 1–10.

Li, XiaoLin, Li, Yupeng, & Wang, Yu. 2016. Minimising makespan on a batch processing machine using heuristics improved by an enumeration scheme. *International Journal of Production Research*, **55**(06), 1–11.

Li, Xueping, & Zhang, Kaike. 2018. Single batch processing machine scheduling with two-dimensional bin packing constraints. *International Journal of Production Economics*, **196**, 113–121.

Mathirajan, M, Sivakumar, AI, & Chandru, Vijaya. 2004. Scheduling algorithms for heterogeneous batch processors with incompatible job-families. *Journal of Intelligent Manufacturing*, **15**(6), 787–803.

Melouk, Sharif, Damodaran, Purushothaman, & Chang, Ping-Yu. 2004. Minimizing makespan for single machine batch processing with non-identical job sizes using simulated annealing. *International Journal of Production Economics*, **87**(2), 141–147.

Ozturk, Onur, Espinouse, Marie-Laure, Di Mascolo, Maria, & Gouin, Alexia. 2011. Makespan minimisation on parallel batch processing machines with non-identical job sizes and release dates. *International Journal of Production Research*, **50**(01), 1–14.

Prins, Christian. 2004a. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & operations research*, **31**(12), 1985–2002.

Prins, Christian. 2004b. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, **31**(12), 1985–2002.

Ram, Bala, & Patel, Gunvant. 1998. Modelling furnace operations using simulation and heuristics. *Pages 957–963 of: Proceedings of the Winter Simulation Conference 1998*, vol. 2. IEEE.

Rivera, Juan, Afsar, H., & Prins, Christian. 2015. A multistart iterated local search for the multitrip cumulative capacitated vehicle routing problem. *Computational Optimization and Applications*, **61**(05).

Rivera, Juan Carlos, Afsar, H Murat, & Prins, Christian. 2013. Multistart evolutionary local search for a disaster relief problem. *Pages 129–141 of: International conference on artificial evolution (evolution artificielle)*. Springer.

Santos, Luís, Coutinho-Rodrigues, João, & Current, John R. 2010. An improved ant colony optimization based algorithm for the capacitated arc routing problem. *Transportation Research Part B: Methodological*, **44**(2), 246–266.

Tang, Ke, Mei, Yi, & Yao, Xin. 2009. Memetic algorithm with extended neighborhood search for capacitated arc routing problems. *IEEE Transactions on Evolutionary Computation*, **13**(5), 1151–1166.

Trindade, Renan Spencer, de Araújo, Olinto César Bassi, Fampa, Marcia Helena Costa, & Müller, Felipe Martins. 2018. Modelling and symmetry breaking in scheduling problems on batch processing machines. *International Journal of Production Research*, **56**(22), 7031–7048.

Trindade, Renan Spencer, de Araújo, Olinto CB, & Fampa, Marcia. 2020. Arc-flow approach for parallel batch processing machine scheduling with non-identical job sizes. *Pages 179–190 of: International Symposium on Combinatorial Optimization*. Springer.

Universidad EAFIT. 2017. *Reglamento de propiedad intelectual.*

Uzsoy, Reha. 1994. Scheduling a single batch processing machine with non-identical job sizes. *The International Journal of Production Research*, **32**(7), 1615–1635.

Van De Rzee, DJ, Van Harten, Aart, & Schuur, Peter Cornelis. 1997. Dynamic job assignment heuristics for multi-server batch operations-a cost based approach. *International Journal of Production Research*, **35**(11), 3063–3094.

Velez-Gallego, Mario C., Damodaran, Purushothaman, & Rodríguez, M. 2011. Makespan minimization on a single batch processing machine with unequal job ready times. *International Journal of Industrial Engineering : Theory Applications and Practice*, **18**(01), 536–546.

Villegas, Juan, Prins, Christian, Prodhon, Caroline, Medaglia, Andrés, & Velasco, N. 2013. A matheuristic for the truck and trailer routing problem. *European Journal of Operational Research*, **230**(10), 231–244.

Villegas, Juan G., Prins, Christian, Prodhon, Caroline, Medaglia, Andrés L., & Velasco, Nubia. 2010. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, **23**(5), 780–794. Advances in metaheuristics for hard optimization: new trends and case studies.

Xu, Rui, Chen, Huaping, & Li, Xueping. 2012. Makespan minimization on single batch-processing machine via ant colony optimization. *Computers & Operations Research*, **39**(3), 582–593.

Yuan, JJ, Liu, ZH, Ng, Chi To, & Cheng, TC Edwin. 2004. The unbounded single machine parallel batch scheduling problem with family jobs and release dates to minimize makespan. *Theoretical Computer Science*, **320**(2-3), 199–212.