Integer Programming Based Methods Applied to Cutting, Packing, and Scheduling

Vinícius L. de Lima[1], **Thiago A. de Queiroz** [2], **Flávio K. Miyazawa** [1]

[1]Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)
Campinas - SP, Brazil

[2]Instituto de Matemática e Tecnologia – Universidade Federal de Catalão (UFCAT)
Catalão - GO, Brazil

{v.loti,fkm}@ic.unicamp.br, taq@ufcat.edu.br

Abstract. *We propose many contributions related to combinatorial optimization. First, we propose a number of exact methods for a general class of integer programming models, allowing applications to several important combinatorial optimization problems. To evaluate the effectiveness of our methods, we apply them to many well-studied cutting, packing, and scheduling problems, including the classical bin packing problem. The proposed methods could solve a large number of open benchmark instances, becoming the leading algorithms and the new state-of-the-art of all these problems. We also propose contributions to facilitate future research on two-dimensional cutting and packing. Lastly, we solve a complex problem arisen from a real-world case study in the food industry.*

1. Introduction

We study solution methods for *combinatorial optimization*, a very general study field found in the intersection of mathematics and computer science [Nemhauser and Wolsey 1988]. In combinatorial optimization problems, we are given a (potentially huge) discrete set of solutions, and we need to determine one solution that optimizes a given objective function. These problems model an extensive number of real-world applications, in particular to logistics and supply chain, including, e.g., cutting, packing, production scheduling, vehicle routing, satellite scheduling, facility location, airline network design.

Most of the practical combinatorial optimization problems are $\mathcal{NP}$-hard, and methods that limit the space for the search of an optimal solution are crucial in practice. A method that guarantees to return an optimal solution is called *exact*, whereas a method that does not have this guarantee is called *heuristic*. In general, the practical success of heuristics is based on guiding the search to quickly find (good-quality) feasible solutions. This aspect is also important in exact methods, but another important aspect in these methods is the ability to eliminate from the search large sets of solutions that are not candidates to improve the best known solution, without the need to verify all of them. This aspect is mainly based on special techniques to compute *bounds* on the optimal solution value. These two kinds of methods typically go together. On the one hand, heuristics are often used in the internal of exact methods to improve the search. On the other hand, although the use of heuristics is preferred in practice, exact methods can easily become heuristics by imposing a time limit on the search.

State-of-the-art methods for many hard problems in combinatorial optimization relies on *Mixed Integer Linear Programming* (MILP). MILP models are based on: a set of variables whose domains are either integer or continuous; a linear objective function; and a set of linear constraints. In general, these models are solved by *branch-and-bound*, in which the bound on the optimal solution value is given by the *linear relaxation* of the model, i.e., a model in which the domain of integer variables is relaxed to be continuous. The strength of the linear relaxation plays a key role in the solution efficiency of MILP models. Strong relaxations are crucial to guide the search for good-quality solutions, and also to improve *fathoming* of branch-and-bound nodes, i.e., to eliminate, without loss of optimality, (potentially large) sets of solutions without the need to verify them.

The most straightforward method to solve MILP models is to use general MILP solvers. These solvers represent a powerful approach to solve models with small or medium size and reasonably strong relaxations. However, when models are either too weak or large, more sophisticated methods are required. The issues caused by weak relaxations may be addressed by relying on reformulations, which often derive models that are much larger. On the other hand, the issues from models that are too large are usually addressed by decomposition techniques. In either case, the direct use of an MILP solver is not a practical approach.

One of the most successful techniques to enhance the strength of MILP models is the reformulation by the Dantzig-Wolfe (DW) decomposition [Dantzig and Wolfe 1961, Vanderbeck and Savelsbergh 2006], which has been the base of the state-of-the-art for many important problems since the sixties. The DW decomposition may derive models that are much stronger but with exponentially more variables. Due to the huge size of the resulting models, their solution is usually based on sophisticated techniques such as *branch-(and-cut-)and-price* algorithms. Nonetheless, with some basic assumptions, models resulting from DW decompositions can be reformulated as network flow models [Vanderbeck 2000], while preserving relaxation strength. Formally, a network flow formulation aims at finding an optimal-cost flow in a network while satisfying side constraints. A particular instance of a network flow formulation is an *arc flow model*, in which decision variables correspond to the flow on individual arcs of the network. With this characterization, an exponential-size model derived from DW decomposition may derive an arc flow model with pseudo-polynomial size and same strength.

Contents

In the thesis, we propose several general methods that could lead to the best exact algorithms for many problems – most of them being very classical and fundamental to computer science and operations research, and that have been investigated for several decades. The thesis is divided into three parts.

In the first part, summarized in Section 2, lies our main methodological contributions, which address the solution of formulations derived from DW decompositions that can be represented as pseudo-polynomial arc flow models with strong relaxations. Although the use of general MILP solvers is a good approach for small- or medium-sized models, hard instances of many practical problems usually derive pseudo-polynomial arc flow models that are too large to be efficiently solved directly by such solvers. This gives rise to one of our main concerns: "how to develop solution methods to efficiently solve large arc flow models with strong relaxations, while still taking advantage of the continuously increasing potential of general MILP solvers?". We answer this question by proposing a number of exact methods to solve such models.

Impact. Our methods are very general and can be applied to a large class of combinatorial optimization problems. To evaluate their practical efficiency, we apply them to a variety of well-studied cutting, packing, and scheduling problems from the literature. The presented computational results showed that the use of these methods could solve a large number of open instances of all problems, setting a new state-of-the-art, and becoming the leading exact algorithms for the following problems:

- BIN PACKING PROBLEM (BPP) - a famous problem fundamental in the areas of computer science, logistics, and operations research. The BPP is theoretically equivalent to the *cutting stock problem* (CSP), another classical problem. These problems allow applications whenever we want to minimize the number of stations to allocate a set of tasks. For instance, there is a huge number of industrial applications in supply chain, production, and logistics [Eliiyi and Eliiyi 2009], but also in computing, as in the allocation of data into memory blocks [American Optimization Society 2022] and in cloud computing [Kumaraswamy and Nair 2019]. The high relevance of the BPP and CSP is also shown by the survey [Delorme et al. 2016] that reviews over 150 references related to their exact methods. Their structure and applications have been studied since the thirties and the first MILP models were presented in the early sixties. Thereafter, a long history of improved exact algorithms have appeared, in particular in the last few years [Delorme and Iori 2020, Wei et al. 2020, Pessoa et al. 2020], making new improvements a very hard task to achieve. Our methods could undoubtedly improve all previous best algorithms;
- TWO-STAGE GUILLOTINE CUTTING STOCK PROBLEM (2GCSP) - one of the most popular guillotine cutting problems, studied since the sixties (c.f. the survey by [Iori et al. 2021]);
- ORDERED OPEN-END BIN PACKING PROBLEM (OOEBPP) - a problem arisen from fare payment in subway stations. State-of-the-art of this problem consists of sophisticated algorithms that have been published in the most renowned journals in the area [Ceselli and Righini 2008];
- SKIVING STOCK PROBLEM (SSP) - a problem with many different important applications in, e.g., industrial production processes, manufacturing and inventory plannings, multiprocessor scheduling problems, and wireless communications [Martinovic et al. 2020];
- PARALLEL MACHINE SCHEDULING PROBLEMS - a very general class of problems that model important applications to production scheduling. We combine our general methods with novel techniques for these problems, and apply them to the variant that minimizes weighted completion times, which has been studied for over half a century [Eastman et al. 1964].

The second part of the thesis, summarized in Section 3, follows with contributions to facilitate future research in an important area of combinatorial optimization: two-dimensional orthogonal cutting and packing. These problems have been studied for many decades, and the last comprehensive review related to them was published two decades ago. Since then, the literature on these problems has grown considerably. For that, we propose an extensive literature review and an online library with useful resources related to two-dimensional cutting and packing problems.

In the third part, summarized in Section 4, we solve a problem arising from a colaboration with a company, related to integrated workforce allocation and scheduling of daily orders. The problem consists of a large set of complex decision variables and constraints. To solve it efficiently we propose a constructive heuristic embedded within a multi-start framework.

2. First Part

The first part studies exact methods for pseudo-polynomial arc flow models. Since the seminal work [Valério de Carvalho 1999] to solve a pseudo-polynomial arc flow model in practice was published, around two decades ago, the popularity of pseudo-polynomial arc flow models has increased a lot, and they have been used to solve a wide variety of important problems. Although these models have been successfully used in practice many times, their theoretical foundations was not discussed in depths.

Major contribution 1. We fulfilled this gap in the literature by proposing a survey, presented in Chapter 2 of the thesis, that reviews over 100 references and presents in details the main theoretical foundations of these formulations, helping to understand their strength and providing several insights that may help future research in this envisaged area. This work has been published as an invited review in the *European Journal of Operational Research* [de Lima et al. 2022].

In the survey, we show a relation between the network of pseudo-polynomial arc flow models and Dynamic Programming (DP), through a link with models obtained by DW decomposition. The relation with DP allows to relate state-space relaxation methods for DP with arc flow models. We also present a dual point of view to contrast the linear relaxation of arc flow models with that of models based on paths and cycles. To conclude, we review the main solution methods and applications of arc flow models based on DP in several domains such as cutting, packing, scheduling, and routing.

Major (and main) contribution 2. Thereafter, in Chapter 3, we propose the *Network Flow Framework* (NF-F), which solves DW decompositions that can be represented as pseudo-polynomial arc flow models. NF-F addresses the issue of huge networks, while focusing on models with very strong relaxations. The framework combines a number of advanced techniques, including, e.g., column generation, reduced-cost variable-fixing (RCVF), and branch-and-price. A preliminary paper describing NF-F has been presented in the *Integer Programming and Combinatorial Optimization* conference [de Lima et al. 2021] and a full paper has been published in *Mathematical Programming* [de Lima et al. 2022]. The proposed techniques are summarized in the following:

- Following the network flow characterization, the adopted column generation algorithm generates complete paths at each pricing iteration. Our main contribution is a pricing algorithm that generates multiple paths, with a substantially small increase in the computational effort, which has shown to be very efficient in practice;
- We adopt RCVF to remove from the network, without loss of optimality, arcs that cannot improve the incumbent solution. In general, RCVF depends on a dual solution of the linear relaxation. Our main contribution in this regard is a new general methodology to find dual solutions that may increase the effectiveness of

RCVF. In practice, these new methodologies helped to solve hard open instances already at the root node of the branch-and-price algorithm;

- Another successful proposal is the branching scheme, which solves the original (large) arc flow model by solving a sequence of much smaller models directly by a MILP solver. This branching scheme is tailored to find good-quality solutions already at the first branches. In practice it has helped to solve many hard open instances by solving very few (and, often, even a single) small arc flow models within a considerably small computational time;
- We also propose many theoretical results, which guarantee the correctness of the proposed practical techniques and also motivates the proposed methodologies for RCVF.

Table 1. Comparison with the state-of-the-art for the BPP/CSP (time limit 3600s)

		Delorme and Iori		Wei et al.		Pessoa et al.		Arc Flow		NF-F	
Class	# ins	time	opt	time	opt	time	opt	time	opt	time	opt
AI200	50	8.5	**50**	4.2	**50**	52.3	**50**	21.5	**50**	2.0	**50**
AI400	50	1205.0	40	398.1	46	491.4	47	904.2	44	25.2	**50**
AI600	50	-	-	1759.6	27	1454.1	35	3326.9	9	192.4	**49**
AI800	50	-	-	2766.3	15	2804.7	28	3600.0	0	566.5	**46**
AI1000	50	-	-	3546.1	2	-	-	3600.0	0	1577.1	**36**
ANI200	50	49.3	**50**	13.9	**50**	16.7	**50**	16.3	**50**	3.0	**50**
ANI400	50	2703.9	17	436.2	47	96.0	**50**	1252.5	42	24.9	**50**
ANI600	50	-	-	3602.7	0	3512.5	3	3473.5	6	140.7	**50**
ANI800	50	-	-	3605.9	0	3600.0	0	3600.0	0	393.2	**49**
ANI1000	50	-	-	3637.7	0	-	-	3600.0	0	1302.5	**43**
Falkenauer T	80	1.0	**80**	1.9	**80**	16.0	**80**	0.8	**80**	0.3	**80**
Falkenauer U	80	0.1	**80**	3.8	**80**	-	-	0.1	**80**	0.1	**80**
Hard	28	4.2	**28**	41.5	**28**	17.0	**28**	39.9	**28**	23.6	**28**
Random	3840	-	-	6.2	**3840**	-	-	1.4	**3840**	0.9	**3840**
Scholl	1210	6.6	**1210**	5.0	**1210**	-	-	8.2	**1210**	1.4	**1210**
Schwerin	200	0.2	**200**	0.3	**200**	-	-	1.3	**200**	0.2	**200**
Waescher	17	41.3	**17**	8.7	**17**	-	-	1510.0	**17**	161.2	**17**
Overall	5955	402.0	1772	1166.9	5692	1206.1	371	1491.5	5641	259.7	**5928**

Applications and main results. NF-F is very general and can be applied to a wide variety of combinatorial optimization problems. To evaluate its practical efficiency, we provide applications to four well-studied cutting and packing problems. The most important application is the BPP (equivalent to the CSP), but we also provide applications to the 2GCSP, the SSP, and the OOEBPP. Table 1 presents a comparison of the BPP/CSP results of the three state-of-the-art methods with those obtained by solving an arc flow model by a state-of-the-art general MILP solver (under column Arc Flow) and by our framework (under column NF-F). We present the results for the 17 main benchmark classes in the literature, comprising 5955 instances. For each method and benchmark class, we present the average computational time (under column time) and the total number of instances solved to optimality (under column opt). For each class, the best opt results are presented in bold. The hardest group of benchmark classes for the BPP/CSP, namely AI and ANI, consist of 500 instances. The pseudo-polynomial model in [Delorme and Iori 2020], and the branch-and-cut-and-price algorithms in [Wei et al. 2020] and [Pessoa et al. 2020] could solve, respectively, 157, 237, and 263 of these instances. NF-F could consistently improve these results by solving 473 of these instances, representing around 80% more instances than the best previous results. In addition, NF-F could solve all instances for the other 7 benchmark classes typically considered in the literature. Besides, NF-F could solve a large number of open instances (and become the leading exact algorithm) also for the other problems we considered, i.e., the 2GCSP, the SSP, and the OOEBPP. This shows that our proposals are powerful both in terms of generality and practical efficiency, representing a very general contribution to the state-of-the-art.

Major contribution 3. To conclude the first part, in Chapter 4, we propose a number of additional exact methods for pseudo-polynomial arc flow models with strong relaxations, but focusing in applications on parallel machine scheduling problems with the objective of minimizing weighted completion times. We propose novel solution methods for general arc flow models, but also a new branching scheme tailored for this class of scheduling problems. The hardest benchmark class for the considered problem consists of 560 instances. The state-of-the-art method [Kramer et al. 2019] could solve 502 of these instances, whereas our algorithm could solve all of the 560 instances to proven optimality, proving the practical effectiveness of our proposals. We are currently working on extending the proposed methods to apply them in other classes of parallel machine scheduling problems. Then, we plan to extend Chapter 4 into a full paper to be submitted to a top-quality international scientific journal.

3. Second Part

The second part of the thesis is devoted to two-dimensional cutting and packing problems. It contains an extensive literature review and an introduction to a new online library. We focus on the four main problems in the literature of orthogonal cutting and packing:

- the two-dimensional strip packing problem: find a packing of minimum height into a single bin with fixed width;
- the two-dimensional bin packing problem: determine the minimum number of bins needed to pack the items;
- the two-dimensional knapsack problem: find a packing of maximum value into a single bin;
- the two-dimensional orthogonal packing problem: find a feasible item packing (if any) into a single bin.

We also consider their most studied variants, which include, e.g., orthogonal rotations, guillotine cuts, variable-sized bins, and loading/unloading constraints.

Major contribution 4. In Chapter 5, we review over 180 references related to solution methods for these problems. To the best of our knowledge, the last extensive review on the same class of problems was presented almost two decades ago, and since then, the literature on this area has grown considerably. In this way, our work represents a meaningful update of the last survey in the area. A result of this contribution is a paper published as an invited review in the *European Journal of Operational Research* [Iori et al. 2021].

In our study on two-dimensional cutting and packing problems, we noticed several issues regarding benchmark classes for theses problems: the number of benchmark classes in the literature is huge, and even instances proposed in the seventies are still used to evaluate new methods; there is no clear correspondence between tested variants and adopted benchmark classes as, due to similarity among some of the variants, the same benchmark classes have been adopted for different variants; many benchmark classes are available in different specific libraries, where they are provided in different formats, not always clearly detailed; some benchmark classes were originally published in personal web pages which no longer exist.

Major contribution 5. To address these issues, we propose a unified format for two-dimensional orthogonal cutting and packing problem instances, and converted the main 25 benchmark classes from the literature (comprising over 3000 instances) to this unified format. Then, we propose an online library (introduced in Chapter 6) containing useful resources for research related to the two-dimensional cutting and packing problems discussed in our survey. The library (available at `http://or.dei.unibo.it/library/2dpacklib`) makes available the benchmark classes in the unified format, provides direct links to surveys and typologies, and includes a list of relevant links. The contents of this chapter has been published as a full paper in *Optimization Letters* [Iori et al. 2022].

4. Third Part

The third part of the thesis, consisting of Chapter 7, is concerned with a problem arising from a collaboration with an Italian company from the food industry. The company is responsible for processing and transporting meat to be distributed to supermarkets over many countries. We studied the process related to the production system of the company, which consists of integrated workforce allocation and scheduling of the daily set of orders. The scheduling part is, in particular, a two-stage flexible flow shop problem that is present in many other real-world scenarios in the industry of perishable products.

Every day, the company receives a set of orders that must be processed and delivered within different due dates throughout the day. There are benches where the meat is prepared (cut), either by automated machines or manual labor. To begin the production of the orders, large pieces of meat are processed in benches into smaller pieces that are sent to conveyors. The conveyor workstations consist of workers which take the processed meat, pack them into disposable trays and put them back in the conveyors. The trays are transported to stamp machines, which stamp a label on each tray. Then, the trays are taken to the storage system, to be later delivered to their destinations.

Several operational constraints must be taken into consideration. Some of them are the following: Orders have release dates and due dates; Each order can be processed only in specific benches and conveyors; Machines (i.e., benches and conveyors) have a minimum and maximum number of operating workers; There are machine setups, subject to the sequence of products to be processed; Some benches are directly connected to conveyors, while others are not; A transportation time between benches and conveyors must be taken into account; There is a maximum time a product can wait before being transported from a bench to a conveyor.

The problem consists of many decision variables. First, we must decide the number of working periods in a workday, their lengths, and the number of workers in each period. In each working period, we must assign a fixed number of workers to operate each machine. This is an important decision, since workers cannot be reallocated. In the scheduling part, we must determine the benches and conveyors to process each order, and also the sequencing of orders to be processed in each machine.

Major Contribution 6. Although two-stage flexible flow shop problems are popular in the literature (for a survey, see [Tian-Soon Lee 2019]), the problem we study has a special set of constraints that, to the best of our knowledge, has never been considered in these problems before. These are the constraints related to the transportation of orders between the first stage (benches) and second stage (conveyors), and the maximum waiting time an order can stay in a buffer between the two stages. These constraints can be found in many applications in the industry of perishable products. In this way, our first contribution is the introduction of this problem, which has a considerably general applicability, to the literature. To solve the integrated problem, we propose a constructive heuristic, which is embedded within a random multi-start framework. The results of our computational experiments over realistic instances show that the heuristic is effective and can support the company on its daily decisions. The algorithm was implemented in the company system, and the company attested that it provides satisfactory results, when compared to their previous solution methods. The contents of Chapter 7 has been presented at the *Advances in Production Management Systems 2021* conference [Bolsi et al. 2021]. We have been invited to submit an extended version of the paper to the special issue of the conference. For that, we submitted a full paper related to an extension of the previously proposed solution that embeds the constructive algorithm within Variable Neighborhood Search and a Biased Random-Key Genetic Algorithm.

5. Final Considerations on our Main Methodological Contributions

One of the main reasons of the increase in popularity of pseudo-polynomial arc flow models is the possibility to solve practical hard problems directly by general MILP solvers. Yet, many hard instances from the real-world produce pseudo-polynomial arc flow models that are too large to be efficiently solved directly by these solvers. Our proposals represent a general methodology to solve such models while still taking advantage of the continuously increasing potential of MILP solvers. Our general framework consists of many novel techniques, of which, we highlight:

- A non-trivial column generation algorithm that can solve the linear relaxation of hard instances under a very limited time;
- A number of variable-fixing strategies that helped solve several hard open benchmark instances already at the root node of the branch-and-price tree;
- A powerful branching scheme that helped finding the optimal solution of many open benchmark instances under a few seconds of computational time;
- A number of theoretical results that establishes the correctness of the proposed techniques and motivates the development of novel methods.

General Applicability. Our experiments proved our methods to be very powerful for several hard optimization problems that have been studied in the literature for several years, some of them being very fundamental and famous to the computer science area:

- Bin packing / cutting stock problem;
- Two-stage guillotine cutting stock problem;
- Skiving stock problem;
- Ordered open-end bin packing problem;

- Parallel machine scheduling problem with minimum weighted completion times.

Generality x practical effectiveness. Our methods, which are very general, covering a wide range of possible applications, not only provide better results than the direct use of a MILP solver, but also became the best exact algorithm for all tested problems, by outperforming their state-of-the-art, which mostly consist of algorithms tailored for the specific problems. This shows that our proposals are powerful both in terms of generality and practical efficiency.

Contributions to existing and future approaches. We believe that not only our complete framework is a valuable contribution, but also that the insights behind its specific components can be adopted in the internal of problem-specific algorithms. This may be a promising direction to improve the state-of-the-art for other hard problems without the need to specifically rely on network flow characterizations or strong relaxations. In this way, we are confident that our proposed methods not only represent a new state-of-the-art for many hard problems, but also may inspire new methodologies to further improve solution methods for other problems in the important field of combinatorial optimization.

Acknowledgments

We thank *The São Paulo Research Foundation* (FAPESP) for funding this research project (grant number 2017/11831-1).

References

American Optimization Society (2022). Bin packing and machine scheduling. https://www.ams.org/publicoutreach/feature-column/fcarc-packings3. Accessed: 2022-03-22.

Bolsi, B., de Lima, V., de Queiroz, T., and Iori, M. (2021). Integrated workforce scheduling and flexible flow shop problem in the meat industry. In Dolgui, A., Bernard, A., Lemoine, D., von Cieminski, G., and Romero, D., editors, *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems*, pages 594–602. Springer.

Ceselli, A. and Righini, G. (2008). An optimization algorithm for the ordered open-end bin-packing problem. *Operations Research*, 56(2):425–436.

Dantzig, G. and Wolfe, P. (1961). The decomposition algorithm for linear programs. *Econometrica*, 29(4):767–778.

de Lima, V., Alves, C., Clautiaux, F., Iori, M., and Valério de Carvalho, J. (2022). Arc flow formulations based on dynamic programming: Theoretical foundations and applications. *European Journal of Operational Research*, 296(1):3–21.

de Lima, V., Iori, M., and Miyazawa, F. (2021). New exact techniques applied to a class of network flow formulations. In Singh, M. and Williamson, D., editors, *Integer Programming and Combinatorial Optimization*, pages 178–192. Springer.

de Lima, V., Iori, M., and Miyazawa, F. (2022). Exact solution of network flow models with strong relaxations. *Mathematical Programming* (forthcoming).

Delorme, M. and Iori, M. (2020). Enhanced pseudo-polynomial formulations for bin packing and cutting stock problems. *INFORMS Journal on Computing*, 32(1):101–119.

Delorme, M., Iori, M., and Martello, S. (2016). Bin packing and cutting stock problems: Mathematical models and exact algorithms. *European Journal of Operational Research*, 255(1):1–20.

Eastman, W., Even, S., and Isaacs, I. (1964). Bounds for the optimal scheduling of n jobs on m processors. *Management Science*, 11(2):268–279.

Eliiyi, U. and Eliiyi, D. (2009). Applications of bin packing models through the supply chain. *International Journal of Business and Management Studies*, 1(1):11–19.

Iori, M., de Lima, V., Martello, S., Miyazawa, F., and Monaci, M. (2021). Exact solution techniques for two-dimensional cutting and packing. *European Journal of Operational Research*, 289(2):399–415.

Iori, M., de Lima, V., Martello, S., and Monaci, M. (2022). 2DPackLib: A two-dimensional cutting and packing library. *Optimization Letters*, 16(2):471–480.

Kramer, A., Dell'Amico, M., and Iori, M. (2019). Enhanced arc-flow formulations to minimize weighted completion time on identical parallel machines. *European Journal of Operational Research*, 275(1):67–79.

Kumaraswamy, S. and Nair, M. (2019). Bin packing algorithms for virtual machine placement in cloud computing: a review. *International Journal of Electrical and Computer Engineering*, 9(1):512.

Martinovic, J., Delorme, M., Iori, M., Scheithauer, G., and Strasdat, N. (2020). Improved flow-based formulations for the skiving stock problem. *Computers & Operations Research*, 113 article 104770.

Nemhauser, G. and Wolsey, L. (1988). *Integer and Combinatorial Optimization*. Wiley & Sons.

Pessoa, A., Sadykov, R., Uchoa, E., and Vanderbeck, F. (2020). A generic exact solver for vehicle routing and related problems. *Mathematical Programming*, 183:483–523.

Tian-Soon Lee, Y.-T. L. (2019). A review of scheduling problem and resolution methods in flexible flow shop. *International Journal of Industrial Engineering Computations*, 10(1):67–88.

Valério de Carvalho, J. (1999). Exact solution of bin-packing problems using column generation and branch-and-bound. *Annals of Operations Research*, 86:629–659.

Vanderbeck, F. (2000). On Dantzig-Wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm. *Operations Research*, 48(1):111–128.

Vanderbeck, F. and Savelsbergh, M. (2006). A generic view of Dantzig–Wolfe decomposition in mixed integer programming. *Operations Research Letters*, 34(3):296–306.

Wei, L., Luo, Z., Baldacci, R., and Lim, A. (2020). A new branch-and-price-and-cut algorithm for one-dimensional bin-packing problems. *INFORMS Journal on Computing*, 32(2):428–443.