A Logic-based Benders Decomposition Approach for the 3-Staged Strip Packing Problem

Johannes Maschler and Günther R. Raidl *

Abstract We consider the 3-staged Strip Packing Problem, in which rectangular items have to be arranged onto a rectangular strip of fixed width, such that the items can be obtained by three stages of guillotine cuts while the required strip height is to be minimized. We propose a new logic-based Benders decomposition with two kinds of Benders cuts and compare it with a compact integer linear programming formulation.

1 Introduction

In the 3-staged Strip Packing Problem (3SPP) we are given n rectangular items and a rectangular strip of width W_S and unlimited height. The aim is to pack the items into the strip using minimal height, s.t. all items can be received by at most three stages of guillotine cuts. In the first stage the strip is cut horizontally from one border to the opposite one and yields up to n levels. In the second stage the levels are cut vertically and at most n stacks are received. In the third stage the stacks are cut again horizontally and the resulting rectangles of the three consecutive stages of guillotine cuts correspond to the n items and the waste.

The general Strip Packing Problem (SPP) was proposed by Baker et al. [1] and has received a large amount of attention: on the one hand many real-world applications, such as glass, paper and steel cutting, can be modeled as SPPs; on the other hand it is strongly-NP hard and has turned out to be a demanding combinatorial problem. We study the special case of the SPP, where only guillotine cuts are al-

Johannes Maschler · Günther R. Raidl
Institute of Computer Graphics and Algorithms, TU Wien, Favoritenstraße 9–11, Vienna, Austria, e-mail: {maschler|raidl}@ac.tuwien.ac.at

* The authors thank Christina Büsing from TU Wien for her support on this paper.

lowed as already considered by Hifi [4] and Lodi et al. [6]. We restrict ourselves to three stages of guillotine cuts. This can be motivated from glass cutting [8].

One of the leading exact approaches for the SPP proposed by Côté et al. [2] is based on a Benders decomposition using combinatorial Benders cuts, which can be seen as an implementation of the logic-based Benders decomposition (LBBD) introduced by Hooker and Ottosson [5]. Their master problem cuts items into unit-width slices and solves a parallel processor problem that requires all slices belonging to the same item have to be packed next to each other. The subproblem consists of transforming a solution of the master problem into a solution of the SPP. However, this algorithm cannot be trivially extended to solve the 3SPP.

In this work we suggest a different form of LBBD specifically for the 3SPP, compare it to a compact integer linear programming (ILP) formulation and show that its performance is competitive. The proposed master problem assigns items to levels and defines the number of stacks in which the items can appear. The resulting subproblems pack items of the same width assigned to the same level into the given number of stacks. Two kinds of Benders cuts are provided, from which the first one are rather straightforward, while the second one are more general.

2 A Logic-based Benders Decomposition for 3SPP

The proposed LBBD consists of a master problem which is a relaxation of the 3SPP. From an optimal solution of the master problem subproblems are derived that yield a complete solution for the 3SPP. If this solution is not yet optimal we improve the master problem with Benders optimality cuts and resolve the master problem.

2.1 Master Problem

The master problem considers n levels and aims to pack the items into these levels. For symmetry breaking item i is only allowed to be packed into level j if $i \geq j$ and only if item j is also packed into level j. This way of symmetry breaking has been already used by Puchinger and Raidl [7]. To better exploit scenarios where many items have the same widths and/or lengths, let us more precisely define the set of all appearing widths as $W = \{w_1, \dots, w_p\}$, the set of all appearing heights as $H = \{h_1, \dots, h_q\}$, and the dimensions of item $i \in I = \{1, \dots, n\}$ as w_{ω_i} and h_{λ_i} with $\omega \in \{1, \dots, p\}^n$ and $\lambda \in \{1, \dots, q\}^n$. We further assume w.l.o.g. that the items and the widths in W are given in a non-decreasing order.

We consider in the master problem different variants of each item, in which depending on a parameter $e \in \{1, 2, \dots\}$ its width is increased by $w_{\omega_i} * e$ while its height is decreased by h_{λ_i}/e. We denote the modified width of an item i by the parameter e with $w_{\omega_i e}$, and analogously, the modified height with $h_{\lambda_i e}$. We require that for each level all packed items of the same width have to be reshaped by the same factor e. The variant of the items models on how many stacks the items meant to be placed. The total height of the packed variant of items with the same width represents the height if the items can be partitioned between the stacks ideally. Therefore, the master problem is indeed a relaxation of the original problem since it models the first two stages but not the third.

The next consideration concerns the maximal number of variants that has to be provided for each item. For a given item, the parameter e can be restricted on the one hand by the width of the strip and on the other hand by the total number of items of that width that can be packed into the considered level. The maximal number of different variants for a level j and a width w_g is

$$e_{\max}(j,g) = \min\left(\left\lfloor \frac{W_\mathrm{S} - w_{\omega_j}}{w_g} \right\rfloor + \begin{cases} 1 & \omega_j = g \\ 0 & \text{otherwise} \end{cases}, \; |\{i = j,\dots,n \,|\, \omega_i = g\}| \right).$$

The first term of the minimum calculates the maximum number of items of width w_g that can be placed next to each other without exceeding W_S. The second term yields the number of items of width w_g that can be packed into level j.

The master problem is modeled by using three sets of variables: Binary variables x_{jie} which are set to 1 iff item i in variant e is assigned to level j. Binary variables y_{jge} which are set to 1 iff an item in variant e with original width w_g is assigned to level j. Integral variables z_j which are set to the height of the corresponding level j. To ease the upcoming notation we denote with $[i,n]$ the set $\{i,\dots,n\}$ and with $E(j,g)$ the set $\{1,\dots,e_{\max}(j,g)\}$. The master problem is defined as follows:

$$\min \sum_{j\in[1,n]} z_j \tag{1}$$

$$\text{s.t.} \sum_{j\in[1,i]} \sum_{e\in E(j,\omega_i)} x_{jie} = 1 \qquad \forall i \in [1,n] \tag{2}$$

$$x_{jie} \le y_{j\omega_i e} \qquad \forall j \in [1,n], \forall i \in [j,n], \forall e \in E(j,\omega_i) \tag{3}$$

$$\sum_{e\in E(j,g)} y_{jge} \le 1 \qquad \forall j \in [1,n], \forall g \in [\omega_j, p] \tag{4}$$

$$\sum_{i\in[j,n]\,|\,\omega_i=g} x_{jie} \ge e\, y_{jge} \qquad \forall j \in [1,n], \forall g \in [\omega_j,p], \forall e \in E(j,g) \tag{5}$$

$$\sum_{e_1\in E(j,\omega_j)} x_{jje_1} \ge x_{jie_2} \qquad \forall j \in [1,n-1], \forall i \in [j+1,n], \forall e_2 \in E(j,\omega_i) \tag{6}$$

$$\sum_{g\in[\omega_j,p]} \sum_{e\in E(j,g)} w_{ge}\, y_{jge} \le W_\mathrm{S} \qquad \forall j \in [1,n] \tag{7}$$

$$h_{\lambda_i} x_{jie} \le z_j \qquad \forall j \in [1,n], \forall i \in [j,n], \forall e \in E(j,\omega_i) \tag{8}$$

$$\sum_{i\in[j,n]\,|\,\omega_i=g} h_{\lambda_i e}\, x_{jie} \le z_j \qquad \forall j \in [1,n], \forall g \in [\omega_j,p], \forall e \in E(j,g) \tag{9}$$

Inequalities (2) force that each item has to be packed in a single variant exactly once. Equations (3) link the variables x_{jie} and $y_{j\omega_i e}$. The restriction that items of the same width have to be packed in the same variant is guaranteed by (4). Inequalities

(5) ensure that items are not meant to be placed on more stacks than the number of items. Constraints (6) impose that items can only be packed into a level j iff also item j is packed into level j. Constraints (7) disallows that the total width of the packed item variants exceed the strip width W_S. Constraints (8) and (9) ensure that the level is at least as high as every single item in it and at least as high as the total height of all packed items of the same width in their corresponding variant.

2.2 Subproblem

The master problem is a relaxation of the 3SPP, since it assumes that items of the same width can be partitioned, s.t. the resulting stacks are of equal height. The subproblems determine the actual packing of the items and with it the actual level height. The resulting subproblems consist of assigning items, of the same width packed by the master into the same level, into the number of stacks determined by their item variant. The aim is the minimize the height of the highest stack. This problem corresponds to the $P||C_{\max}$ problem [3]. We use for the subproblem a straightforward ILP formulation which was solved in less than a second in all considered instances.

2.3 Benders Cuts

The aim of Benders cuts is to incorporate the knowledge obtained in the subproblems back into the master problem. In the simple case a Benders cut states that if a set of items is packed in a certain variant, then the height of the level is at least as high as the result of the corresponding subproblem.

Let $\bar{I}$ and $\bar{e}$ be the set of items and their variant that have defined a subproblem and let $\bar{z}$ be the objective value of a corresponding optimal solution. Moreover, the set J' contains those levels that allow an assignment of items, s.t. the Benders cut can get activated. The simple version of our Benders cuts is

$$\left(\sum_{i \in \bar{I}} x_{ji\bar{e}} - |\bar{I}| + 1\right) \bar{z} \leq z_j \qquad \forall j \in J'. \tag{10}$$

These Benders cuts have the disadvantage that they do not affect item assignments differing from I' only in that items are exchanged by congruent items, i.e., items having the same width and height. The extended Benders cuts aim to overcome this drawback. Let $\bar{H} \subseteq H$ be the set of heights of the items from a subproblem defined by $\bar{I}$ and $\bar{e}$. We introduce for each height $h \in \bar{H}$ a binary variable u_h which is set to true iff at least as many items of the same width, height and variant are packed into a level as it has been packed in the considered subproblem. The set $I' \subseteq I$ contains all items having the corresponding width and height. The constraints that set the u_h

variables are given by

$$\sum_{i \in I'} x_{ji\bar{e}} - |\bar{I}| + 1 \leq u_h(|I'| - |\bar{I}| + 1) \qquad \forall h \in \bar{H},\ \forall j \in J'. \tag{11}$$

The extended Benders cuts impose that the height of a level is at least as high as in the subproblem if all corresponding u_h variables are set to 1 and is defined as

$$\Big(\sum_{h \in \bar{H}} u_h - |\bar{H}| + 1\Big)\bar{z} \leq z_j \qquad \forall j \in J'. \tag{12}$$

Moreover, we iteratively exclude the smallest item of $\bar{I}$ and resolve the subproblem as long as the objective of the optimal solution does not change. The resulting Benders cuts are in general stronger and reduce the number of master iterations.

3 Compact Formulation

The used compact formulation is straightforward, hence we omit an exact specification. The main idea is to pack items first into stacks and then pack stacks into levels. To model this, we use binary variables v_{ki} that are set to one if item i is packed into strip k and binary variables u_{jk} to express that stack k resides in level j. Each item has to be packed exactly once and each stack containing items is allowed to appear in exactly one level. Moreover, we have to ensure that the total width of all stacks belonging to the same level does not exceed W_S. For each of the potentially n levels an integer variable is used which has to be at least as high as the highest residing stack. Furthermore, we applied the symmetry breaking described in Section 2.1.

4 Computational Results

The algorithms have been implemented in C++ and tested on an Intel Xeon E5-2630 v2, 2.60 GHz using Ubuntu 14.04. The ILP formulations have been solved with IBM Ilog Cplex 12.6.2 using the same parameter setting as in [2]. All algorithms had a time limit of 7200 seconds. For the benchmark we use the instance sets beng, cgcut, gcut, ht and ngcut from [2].

We compare the compact formulation against our LBBD with simple Benders cuts and with extended Benders cuts. The compact model could solve 31 out of 47 test instances to optimality, which is only marginally outperformed with 32 optimally solved test instances by the LBBD using either simple or extended Benders cuts. However, the LBBD can solve some instances considerably faster as Figure 1 shows. For instance, after 10 seconds the LBBD with simple and with extended Benders cuts could solve 22 test instances to optimality, while the compact model could optimally solve 17 test instances.

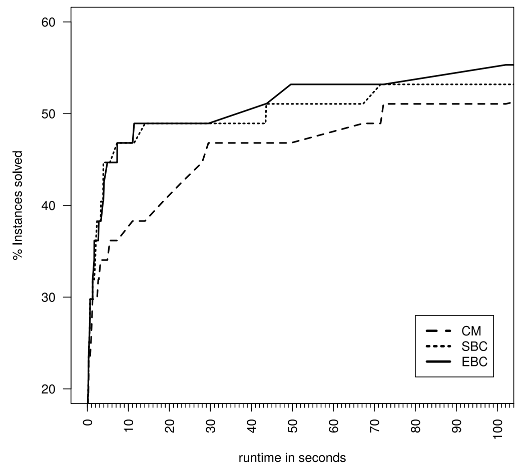

Fig. 1 Performance profile of the first 100 seconds for the compact formulation (CM) and for the presented LBBD with simple Benders cuts (SBC) and extended Benders cuts (EBC) on the instance sets beng, cgcut, gcut, ht and ngcut from [2].

5 Conclusion

We proposed a novel LBBD for the 3SPP and compared it with a compact formulation. The master problem relaxes the 3SPP s.t. only the first two stages of guillotine cuts are determined. The resulting subproblems are iteratively resolved to strengthen the generated Benders cuts. In addition, we proposed two kinds of Benders cuts. The experimental results have shown that the presented LBBD can solve substantially more test instances in the first 100 seconds. The LBBD can solve one test instance more than the compact model within the time limit. More testing is necessary to see under which conditions the proposed approach works especially well.

References

1. Baker, B.S., Coffman Jr, E.G., Rivest, R.L.: Orthogonal packings in two dimensions. SIAM Journal on Computing **9**(4), 846–855 (1980)
2. Côté, J.F., Dell'Amico, M., Iori, M.: Combinatorial benders' cuts for the strip packing problem. Operations Research **62**(3), 643–661 (2014)
3. Graham, R.L., Lawler, E.L., Lenstra, J.K., Kan, A.R.: Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of discrete mathematics **5**, 287–326 (1979)
4. Hifi, M.: Exact algorithms for the guillotine strip cutting/packing problem. Computers & Operations Research **25**(11), 925–940 (1998)
5. Hooker, J.N., Ottosson, G.: Logic-based benders decomposition. Mathematical Programming **96**(1), 33–60 (2003)
6. Lodi, A., Martello, S., Vigo, D.: Models and bounds for two-dimensional level packing problems. Journal of Combinatorial Optimization **8**(3), 363–379 (2004)
7. Puchinger, J., Raidl, G.R.: Models and algorithms for three-stage two-dimensional bin packing. European Journal of Operational Research **183**(3), 1304–1327 (2007)
8. Puchinger, J., Raidl, G.R., Koller, G.: Solving a real-world glass cutting problem. In: Evolutionary Computation in Combinatorial Optimization 2004, pp. 162–173. Springer (2004)