

From Time Representation in Scheduling to the Solution of Strip Packing Problems

Pedro M. Castro,^{†‡} and Ignacio E. Grossmann[‡]*

[†] Unidade de Modelação e Optimização de Sistemas Energéticos, Laboratório Nacional de Energia e Geologia, 1649-038 Lisboa, Portugal

[‡] Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Abstract

We propose two mixed-integer linear programming based approaches for the 2-D orthogonal strip packing problem. Using knowledge from the alternative forms of time representation in scheduling formulations, we show how to efficiently combine three different concepts into the x - and y -dimensions. One model features a discrete representation on the x -axis (strip width) and a continuous representation with general precedence variables on the y -axis (strip height). The other features a full continuous-space representation with the same approach for the y -axis and a single non-uniform grid made up of slots for the x -axis. Through the solution of a set of twenty nine instances from the literature, we show that the former is a better approach, even when compared to three alternative MILP models ranging from a pure discrete-space to a pure continuous-space with precedence variables in both dimensions. All models are available in www.minlp.org.

Keywords: Optimization; Integer programming; Search algorithm; Event points.

* Corresponding author. Tel.: +351-210924643. E-mail: pedro.castro@lneg.pt.

1. Introduction

Time representation is perhaps the most important classification criterion for scheduling models. Mathematical formulations can either be classified as discrete- or continuous-time and several different alternatives have been proposed. Overall, there are essentially four main concepts being used, which are illustrated in Figure 1.

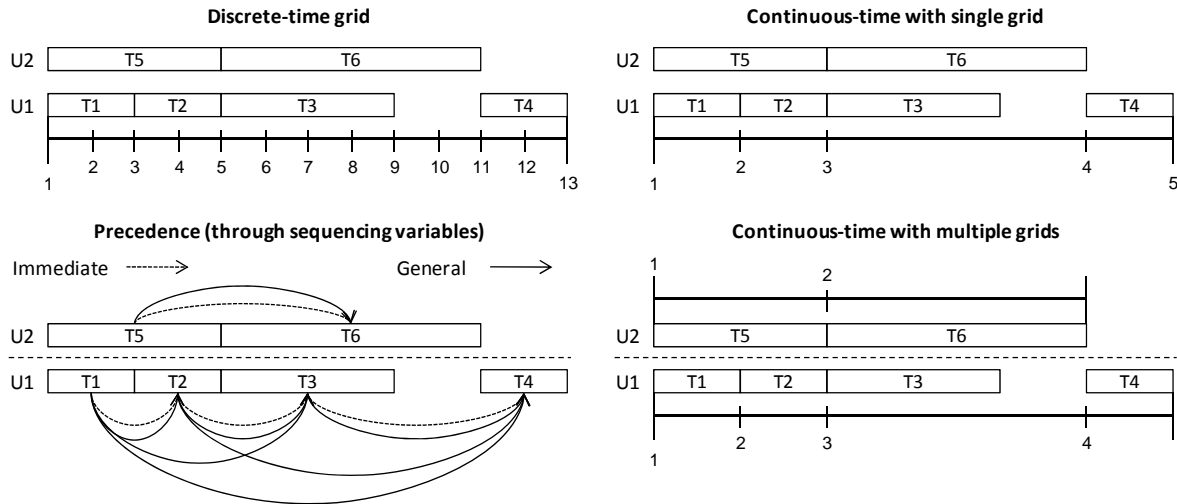


Figure 1. Time representation alternatives in scheduling formulations.

Table 1. Classification of scheduling approaches according to time representation

Discrete-time	Continuous-time		
	Single grid	Multiple grids	Precedence
Kondili et al. (1993)	Zhang & Sargent (1996)	Pinto & Grossmann (1995)	Méndez et al. (2000)
Shah et al. (1993)	Schilling & Pantelides (1996)	Iearapetritou & Floudas (1998)	Méndez et al. (2001)
Pantelides (1994)	Castro et al. (2001)	Giannelos & Georgiadis (2002)	Harjunkoski & Grossmann (2002)
Glismann & Gruhn (2001)	Maravelias & Grossmann (2003b)	Castro & Grossmann (2005)	Gupta & Karimi (2003)
Maravelias & Grossmann (2003a)	Castro et al. (2004)	Liu & Karimi (2007)	Prasad & Maravelias (2008)
Castro et al. (2008)	Sundaramoorthy & Karimi (2005)	Castro & Novais (2008)	Sundaramoorthy & Maravelias (2008)
Sundaramoorthy & Maravelias (2011a)	Giménez et al. (2009)	Shaik & Floudas (2009)	Ferrer-Nadal et al. (2008)
Wassick & Ferrio (2011)	Castro et al. (2009)	Susarla & Karimi (2010)	Capón-Garcia et al. (2009)

Table 1 provides a list of some of the most important scheduling references in the Process Systems Engineering literature in the last 20 years, classified according to the time representation concept

used. It is apparent that no particular option is becoming more common, clearly reflecting that the best option for a problem is very much dependent on its features. While predicting the best performer is often difficult, some guidelines can be given.

The discrete-time approach is perhaps the most powerful and has been shown capable of handling problems of industrial relevance (Wassick, 2009; Wassick & Ferrio, 2011). The time horizon of interest is divided into a fixed number of time slots of predetermined duration, with one knowing a priori the location of all time points. This makes it straightforward to handle holding and backlog costs (Sundaramoorthy & Maravelias, 2011b), thus allowing for easy integration with the higher level planning model (Maravelias & Sung, 2009), intermediate events such as release/due dates, equipment maintenance as well as time-dependent utility pricing and availability (Castro et al., 2009, 2011). On the downside, fixed processing times need to be assumed and approximated to a multiple of the interval length.

Continuous-time models, on the other hand, are more accurate and sensitive to small changes in the duration of processing and changeover tasks, which can be of a different order of magnitude. They are thus more appropriate for integration with the lower level control layer (Capón-García et al. 2011). Deciding for a continuous-time model needs to be followed by the choice of the concept used to keep track of events taking place.

Precedence based models (Méndez et al. 2000, 2001) were the first to appear and are known for their ability to provide high quality solutions with limited computational resources, even though it may be difficult to prove optimality. The concept of general precedence is used more frequently when compared to immediate precedence since it gives rise to smaller models that typically perform better. Precedence based models tend to be less general than their time grid counterparts and are thus more commonly found for multistage plants, where they are more efficient.

In facilities with a network structure involving resource constraints other than equipment and unit availability, i.e. multipurpose plants, time grid based models become the only option. Due to process complexity they are linked to unified frameworks for process representation, the State-Task (Kondili et al. 1993) and Resource-Task (Pantelides, 1994) Networks. When compared to the discrete-time representation, the time horizon is also divided into a fixed number of slots but now the grid(s) is/are

non-uniform with the duration of the slots being determined through a set of continuous variables. Fewer slots are required to represent the solution, more so if multiple time grids are employed. However, the number of prespecified slots has also a stronger influence on both solution quality and computational effort, with one typically requiring an iterative search procedure to find the global optimal solution (Méndez et al., 2006). While it is easier to rely on a single grid, such option forces allowing batch tasks to spread across multiple slots, which severely compromises their performance. With multiple time grids, more efficient unit-specific models can be used even though there is still no general model of such type that can handle all different types of resources and storage policies.

1.1. Going multidimensional

In the context of N-dimensional allocation problems (Westerlund, 2005), scheduling problems can be viewed as one dimensional, with the relevant dimension being time. Strip packing problems are a class of 2-dimensional allocation problems (Amossen & Pisinger, 2010) that are open dimensional (Wascher et al. 2007), meaning that all items need to be packed into a strip of a given width so as to minimize its height. In the paper industry for example, one goal is cutting jumbo reels of paper into smaller reels so as to minimize trim losses. Being a simple to define, albeit challenging problem of industrial relevance, is probably the reason it has been widely studied by the research community and several solution approaches have been proposed. Depending mostly on problem size and the time available to generate a solution, one may rely on heuristics (Wei et al. 2009; Ortmann et al. 2010), exact algorithms (Kenmochi et al. 2009; Martello et al. 2003; Bekrar et al. 2007; Alvarez-Valdes et al. 2009; Grancolas & Pinto, 2010) or mathematical programming models (Castro & Oliveira, 2011). The latter two have the advantage of establishing if the best found solution is indeed optimal while informing of the maximum possible distance to such optimum, which can be quite relevant. The drawback is their sharp decrease in performance with respect to problem size. Mathematical programming approaches are more adaptable to changes in the problem constraints or objective function, an essential feature considering that industrial problems rarely fall within the exact boundaries of a standard problem definition. For instance, Dow Chemical has recently reported (Wassick & Ferrio, 2011) the solution of a non-classical packing problem consisting on loading a

semi-trailer with packages of different sizes and weights so that the trailer payload weight and also the rear axle weight are kept below highway allowable limits. The solution method was a mixed-integer linear programming (MILP) model that extended a discrete-time scheduling model (Wassick, 2009) from one to two dimensions.

The recently proposed MILP-based approaches by Castro and Oliveira (2011) for 2-D packing problems were also inspired by scheduling models. The main novelty has been the combination of two different concepts, a discrete-space representation with respect to the x -axis and a continuous-space representation for the y -axis. Combining discrete with continuous representations is known to be an effective approach for planning and scheduling with a rolling-horizon strategy (Dimitriadis et al. 1997; Castro et al. 2011). Like in scheduling, continuous-space representations for 2-D packing are more accurate since the rectangles widths and heights do not need to be multiples of the characteristic dimensions of the mesh used to discretize the strip, see Figure 2 on the left. Classical MILP models for 2-D packing problems feature continuous variables to determine the (x,y) coordinates of a given rectangle point in the strip (see Figure 2 on the right), a concept that can naturally be extended to three (Wu et al. 2010) and higher dimensional allocation problems (Westerlund et al. 2007). They can be derived from a Generalized Disjunctive Programming formulation featuring four sets of Boolean variables to identify the location of a rectangle with respect to another, followed by either a convex-hull or big-M reformulation (Sawaya & Grossmann, 2005). A closely related model using the centroid coordinate instead of the upper-left corner as the reference point was proposed by Castillo et al. (2005) and reported to have a better computational performance so it is the one considered in this article for comparative purposes.

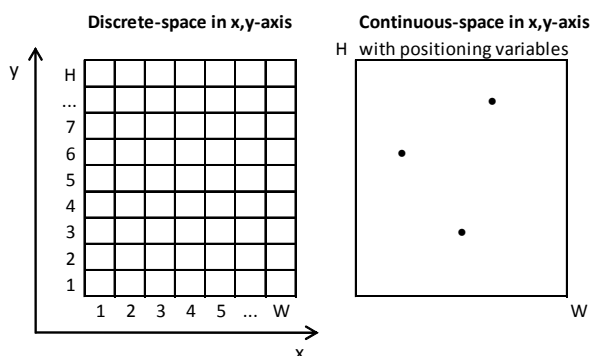


Figure 2. Classical representation alternatives for 2-D allocation problems.

The binary positioning variables in the continuous-space models are the N-dimensional equivalents to the general precedence sequencing variables in scheduling. Models relying on this concept can be used as standalone procedures whereas the solution from those relying on one or multiple spatial grids is dependent on the number of slots specified, and thus requires an iterative search procedure to find the global optimal solution, which might terminate with a suboptimal solution due to a temporary plateau in the value of the objective function. To overcome this limitation, we now propose a new hybrid discrete/continuous-space model featuring sequencing variables in the y -axis, which can not only guarantee global optimality but is also able to improve the performance of the recent hybrid model of Castro & Oliveira (2011). We also present a full continuous-space model that relies on the same set of variables for the y -axis and on a single grid for the x -axis. We have therefore combined all four major concepts for spatial representation given in Figure 1, and efficiently as will be shown later on.

The rest of the paper is structured as follows. Given the problem definition in section 2, we discuss the new 2-D representation alternatives in section 3. The continuous-space model of Castillo et al. (2005) that is part of the computational studies is briefly described in section 4, while the new MILP models are presented next. Section 5 concerns the new hybrid discrete/continuous-space approach with section 6 dealing with the new continuous-space approach. The computational results are then the subject of section 7. In section 8, we summarize the advantages and limitations of the different spatial representation concepts studied with the conclusions being left for section 9.

2. Problem Definition

We consider the two-dimensional orthogonal strip packing problem. Given are a set of rectangles $i \in I$ with width w_i and height h_i . The objective is to place the rectangles without overlap into a strip of a given width W so as to minimize the height H . The focus is set on the solution of problems with integer data and no rotations are allowed. Nevertheless, the continuous-space models that are presented next, naturally handle real values for the widths and heights.

3. Previous and Newly Proposed Spatial Representation Alternatives

In the hybrid model of Castro & Oliveira (2011), the strip is divided along its width W (x -axis) into vertical strips of unitary width. Each elementary strip has its own vertical grid made of allocation points that can be linked to the left-bottom corner of a particular rectangle. All W grids have the same number of allocation points but their y -coordinate will typically be different unless adjacent elementary strips share the same rectangle. This is illustrated on the left of Figure 3.

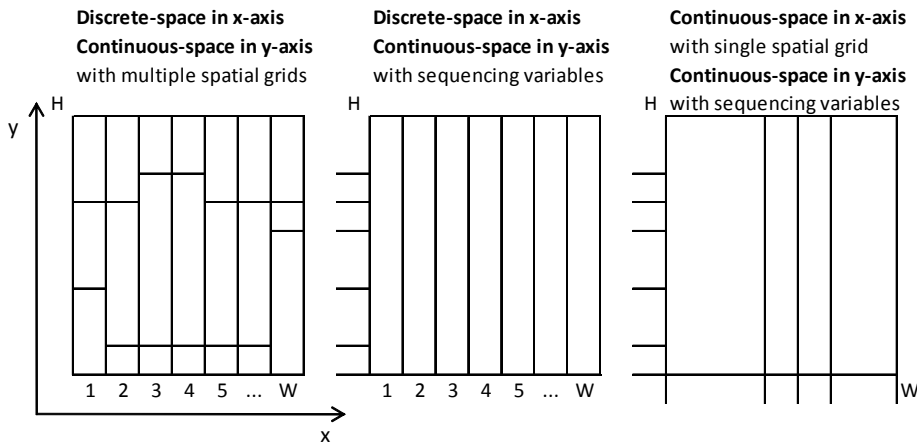


Figure 3. New representation alternatives for 2-D packing problems.

The newly proposed hybrid discrete/continuous-space model uses sequencing variables instead of spatial grids to properly place rectangles with respect to the y -axis. Continuous variables give the y -coordinate of the bottom-edge of the different rectangles with the possibility of the same coordinate being shared by two rectangles provided there is enough horizontal space among them. Figure 3 on the middle illustrates this aspect showing a direct correspondence to the placement of the allocation points from the other hybrid model (Castro & Oliveira, 2011) that are shown on the left (horizontal segments). This novel approach is not cursed by the uncertainty in the number of allocation points meaning that the solution of a single MILP will give the global optimal solution to the problem.

We also propose a somewhat related hybrid continuous-space model that keeps the positioning variables in the y -axis, but uses a single horizontal continuous grid instead of a discrete spatial grid. We thus return to the use of allocation points and the need for an iterative search procedure to find the global optimal solution. This alternative is illustrated on the right of Figure 3. Note that

rectangles can span over multiple slots so the vertical lines do not necessarily correspond to guillotine cuts.

4. A Continuous-Space Model with Positioning Variables (CS)

Two-dimensional continuous-space models typically employ (x_i, y_i) variables to define the rectangle coordinates on the x and y -axis. In order to provide a comparison with the newly proposed models, we consider the adaptation of the BLDP1 block layout design model by Castillo et al. (2005) for the strip packing problem. The big-M formulation by Sawaya and Grossmann (2005) is similar but involves four sets of binary variables in the no-overlap constraints, where at most one can be active.

The objective is the minimization of the strip height, eq. (1). Continuous variables (x_i, y_i) denote the coordinates of the center of rectangle i , while binary variables $P_{i,i'}$ and $Q_{i,i'}$ are sets of binary variables (Westerlund et al. 2007) used in the overlap prevention constraints, eqs. (2-5). Specifically, (2) and (3) are concerned with valid placement of rectangle i to the right and left of rectangle i' , while (4) and (5) ensure valid placement of rectangle i on top or below rectangle i' . Eqs. (6-9) ensure that the edges of every rectangle are within the boundaries of the strip, while the domain of the model variables is given by (10-13).

$$\min H \tag{1}$$

$$\frac{1}{2}(w_i + w_{i'}) \leq x_i - x_{i'} + W \cdot (P_{i,i'} + Q_{i,i'}) \quad \forall i \in I, i' \in I, i' > i \tag{2}$$

$$\frac{1}{2}(w_i + w_{i'}) \leq x_{i'} - x_i + W \cdot (1 + P_{i,i'} - Q_{i,i'}) \quad \forall i \in I, i' \in I, i' > i \tag{3}$$

$$\frac{1}{2}(h_i + h_{i'}) \leq y_i - y_{i'} + M \cdot (1 - P_{i,i'} + Q_{i,i'}) \quad \forall i \in I, i' \in I, i' > i \tag{4}$$

$$\frac{1}{2}(h_i + h_{i'}) \leq y_{i'} - y_i + M \cdot (2 - P_{i,i'} - Q_{i,i'}) \quad \forall i \in I, i' \in I, i' > i \tag{5}$$

$$x_i \geq \frac{1}{2} w_i \quad \forall i \in I \tag{6}$$

$$x_i \leq W - \frac{1}{2} w_i \quad \forall i \in I \tag{7}$$

$$y_i \geq \frac{1}{2}h_i \quad \forall i \in I \quad (8)$$

$$y_i \leq H - \frac{1}{2}h_i \quad \forall i \in I \quad (9)$$

$$P_{i,i'} \in \{0,1\} \quad \forall i \in I, i' \in I, i' > i \quad (10)$$

$$Q_{i,i'} \in \{0,1\} \quad \forall i \in I, i' \in I, i' > i \quad (11)$$

$$x_i \geq 0 \quad \forall i \in I \quad (12)$$

$$y_i \geq 0 \quad \forall i \in I \quad (13)$$

5. New Hybrid Discrete-Space Approach (NDCS)

We now propose a new model that is discrete in the x -axis domain and continuous in the y -axis. The y dimension is preferred for continuous representation to avoid the iterative procedure involved in the efficient determination of the strip height (Castro & Oliveira, 2011), which can now be defined as a continuous variable, H . Let $X = \{1, \dots, W\}$ be the set of vertical slots in the x -axis. Two sets of binary variables are used: $N_{i,x}$ to identify the assignment of the left-edge of rectangle i to the start of slot x ; $Z_{i,i'}$ to specify if the top-edge of rectangle i is below the bottom-edge of rectangle i' ($i' > i$), provided that one box is at least partly above the other, see Figure 4 and Figure 5. The y -coordinate of the bottom-edge of rectangle i is given by variable Y_i .

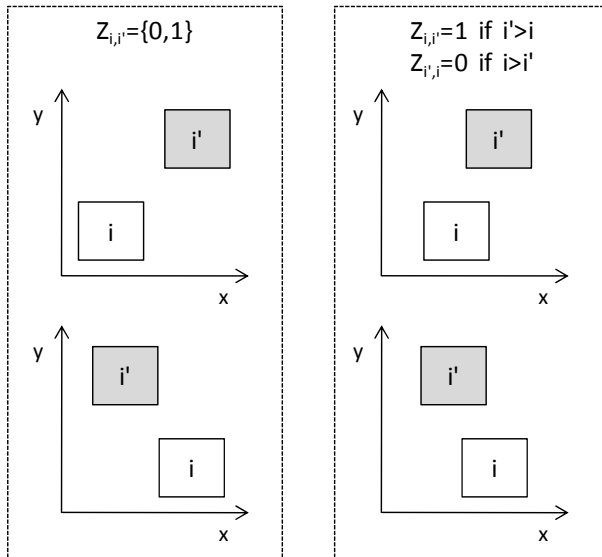


Figure 4. Value of y -axis sequencing variables $Z_{i,i'}$ for different arrangements.

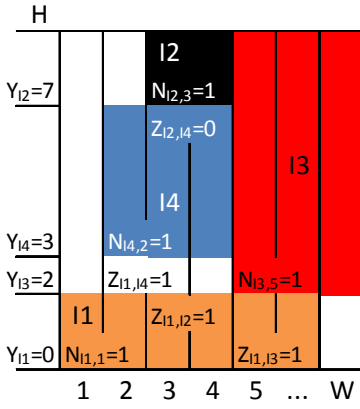


Figure 5. Values of the variables for the new mixed discrete-space approach (NDCS).

Consider the example in Figure 5 that identifies which variables $Z_{i,i'}$ need to be fixed for the arrangement shown. Rectangle I1 is beneath the others and so $Z_{11,12} = Z_{11,13} = Z_{11,14} = 1$. Rectangle I4 is below I2, which is the same as saying that I2 is above I4, i.e. $Z_{12,14} = 0$. With respect to the interaction between (I2,I3) and (I3,I4), they do not occupy the same horizontal slots and so the values of the corresponding variables are irrelevant. This is apparent from big-M constraints (14-15), which are written for every pair of rectangles and every slot x . If the left edge assignments of rectangles i and i' make them occupy slot x , then both summations inside the brackets on the right-hand side will be equal to 1. Then, there are two possibilities: $Z_{i,i'} = 1$ or 0. With the former, the big-M term in eq. (14) disappears and the constraint $Y_{i'} \geq Y_i + h_i$ is enforced, i.e. the bottom-edge of rectangle i' is above the top-edge of rectangle i , as intended by the definition of the binary variables. For the latter option, it is the big-M term in eq. (15) that is equal to zero leading to $Y_i \geq Y_{i'} + h_{i'}$, i.e. rectangle i is above i' . For all other cases the constraints are relaxed.

$$Y_{i'} \geq Y_i + h_i - M \cdot (3 - Z_{i,i'} - \sum_{\substack{x' \in X \\ x-w_i+1 \leq x' \leq x \\ i \in I_{x'}}} N_{i,x'} - \sum_{\substack{x' \in X \\ x-w_{i'}+1 \leq x' \leq x \\ i' \in I_{x'}}} N_{i',x'}) \quad \forall x \in X, i \in I, i' \in I, i' > i \quad (14)$$

$$Y_i \geq Y_{i'} + h_{i'} - M \cdot (2 + Z_{i,i'} - \sum_{\substack{x' \in X \\ x-w_i+1 \leq x' \leq x \\ i \in I_{x'}}} N_{i,x'} - \sum_{\substack{x' \in X \\ x-w_{i'}+1 \leq x' \leq x \\ i' \in I_{x'}}} N_{i',x'}) \quad \forall x \in X, i \in I, i' \in I, i' > i \quad (15)$$

Eq. (16) ensures that the top edge of every rectangle is below the strip height. The same applies to the sum of the heights of all rectangles occupying a given slot x , eq. (17). Then, any given rectangle can be assigned to a single slot, eq. (18). Eqs. (19-22) give the domain of the model variables, while the objective function in eq. (1) also applies.

$$Y_i + h_i \leq H \quad \forall i \in I \quad (16)$$

$$\sum_{i \in I} \sum_{\substack{x' \in X \\ x-w_i+1 \leq x' \leq x \\ i \in I_{x'}}} h_i \cdot N_{i,x'} \leq H \quad \forall x \in X \quad (17)$$

$$\sum_{\substack{x \in X \\ i \in I_x}} N_{i,x} = 1 \quad \forall i \in I \quad (18)$$

$$I_x = \{i \in I : x + w_i \leq W + 1\} \quad \forall x \in X \quad (19)$$

$$N_{i,x} \in \{0,1\} \quad \forall x \in X, i \in I_x \quad (20)$$

$$Z_{i,i'} \in \{0,1\} \quad \forall i \in I, i' \in I, i' > i \quad (21)$$

$$Y_i \geq 0 \quad \forall i \in I \quad (22)$$

6. New Continuous-Space Approach (NCS)

Using the same set of sequencing variables $Z_{i,i'}$, we now propose a new continuous-space model that uses a single spatial grid in the x -axis. Let $E = \{e_1, \dots, e_{|E|}\}$ be the set of allocation points in the x -axis that implicitly fixes the maximum number of rectangles that can be placed on a given horizontal slice of the strip to $|E|-1$, see Figure 6. Rectangles will occupy one or more slots, with binary variables $\bar{N}_{i,e,e'}$ identifying the assignment of the left-edge of rectangle i to allocation point e and the right-edge to point e' ($e' > e$). The x -coordinate of allocation point e will be given by continuous variables X_e .

Using the same simple example, we illustrate the value of the model variables in Figure 7. Notice that rectangle I2 occupies a single slot, starting at allocation point e_3 and ending at allocation point e_4 , thus $\bar{N}_{12,3,4} = 1$. The same goes for I3 while I4 occupies 2 slots and rectangle I1, 4. It is important to highlight that the right edge of the rectangle may not coincide with the x -coordinate of the ending allocation point (e.g. e_5). This avoids the need for further allocation points in cases where the empty space cannot be occupied and is reflected in eqs. (23-24). It states that the difference in the x -coordinates of allocation points e' (replaced by the strip width W if $e' = |E|$, eq. 24) and e must be greater than the width of the rectangle i , provided that its left and right edges are assigned to such points. Notice that the coordinate of the first allocation point is equal to zero, eq (25).

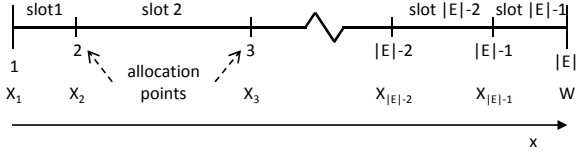


Figure 6. Single non-uniform spatial grid in the x-axis used by new continuous-space approach

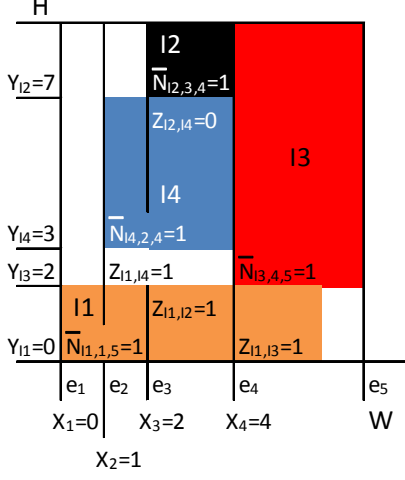


Figure 7. Values of the variables for the new continuous-space approach (NCS).

$$X_{e'} - X_e \geq w_i \cdot \bar{N}_{i,e,e'} \quad \forall i \in I, e \in E, e' \in E, e' > e, e' \neq |E| \quad (23)$$

$$W - X_e \geq w_i \cdot \bar{N}_{i,e,e'} \quad \forall i \in I, e \in E, e' = |E|, e' > e \quad (24)$$

$$X_1 = 0 \quad (25)$$

The other sets of constraints are conceptually similar to those of the hybrid discrete/continuous-space model considering that index x has been replaced by index e . In Eq. (26), the big-M term is equal to zero whenever the top-edge of rectangle i is below the bottom-edge of rectangle i' and both i and i' occupy slot e of the x -domain grid, meaning that their left-edges are allocated to some allocation point e' matching or located to the left of allocation point e and that their right-edges are assigned to some point e'' that is located to the right of both allocation points e and e' . If, on the other-hand, rectangle i is above i' then it is eq. (27) that is enforced.

$$Y_{i'} \geq Y_i + h_i - M \cdot [3 - Z_{i,i'} - \sum_{\substack{e' \in E, e'' \in E \\ e \geq e', e'' > e \\ e'' > e'}} (\bar{N}_{i,e',e''} + \bar{N}_{i',e',e''})] \quad \forall e \in E, e \neq |E|, i \in I, i' \in I, i' > i \quad (26)$$

$$Y_i \geq Y_{i'} + h_{i'} - M \cdot [2 + Z_{i,i'} - \sum_{\substack{e' \in E, e'' \in E \\ e \geq e', e'' > e \\ e'' > e'}} (\bar{N}_{i,e',e''} + \bar{N}_{i',e',e''})] \quad \forall e \in E, e \neq |E|, i \in I, i' \in I, i' > i \quad (27)$$

Eqs. (1), (16), (21-22) are shared with NDCS, while eq. (28) states that the total height of all rectangles occupying a given slot e must be lower than the strip height. Eq. (29) states that the left and right edges of every rectangle i must be assigned to exactly one pair of allocation points (e, e') with $e' > e$. The domain of the new set of binary variables is given by eq. (30). Notice that contrary to NDCS, there can be no domain reduction for the left-edge allocation point.

$$\sum_{i \in I} \sum_{\substack{e' \in E \\ e'' \in E \\ e \geq e' \\ e'' > e'}} h_i \cdot \bar{N}_{i, e', e''} \leq H \quad \forall e \in E, e \neq |E| \quad (28)$$

$$\sum_{e \in E} \sum_{\substack{e' \in E \\ e' > e}} \bar{N}_{i, e, e'} = 1 \quad \forall i \in I \quad (29)$$

$$\bar{N}_{i, e, e'} \in \{0, 1\} \quad \forall i \in I, e \in E, e' \in E, e' > e \quad (30)$$

6.1. Heuristic search for the global optimal solution

The new continuous-space model can find the optimal solution for a given number of allocation points. With a minimum of two points, i.e. one slot, a feasible solution with $H = \sum_i h_i$ can be obtained by packing all rectangles on top of each other, provided that the big-M value in eqs. (26-27) is greater than H . In contrast, the hybrid discrete/continuous-space model in (Castro & Oliveira, 2011) normally requires a few allocation points to ensure feasibility.

In general however, two allocation points will be insufficient to find the real optimal solution to the problem and a higher value will need to be specified. The difficulty is that there is no exact method to predict the value to use, even for one-dimensional scheduling problems and despite recent efforts (Li & Floudas, 2010). This is a serious issue considering that one typically gets a one order of magnitude increase in computational effort following a single increase in the number of allocation points, i.e. events (see Castro & co-workers 2004, 2008, 2011). Since global optimality can only be ensured in the limit of $|E|=|I|+1$, which will almost certainly compromise tractability, a proper search procedure is needed.

We adapt the iterative search procedure from (Castro & Oliveira, 2011) that is in turn borrowed from event-based scheduling models (Méndez et al. 2006). Starting with $|E|=2$, we keep increasing the number of allocation points and solving the optimization problem until the objective function

stops improving. Since the solution space for $|E|+1$ points includes the feasible space for $|E|$, and to enhance computational performance, we remove the current best solution from the feasible space through the use of a cutoff value, see eq. (31). With the purpose of solely searching for a better solution, we make the problem infeasible if the global optimal solution has already been found. One can also specify an absolute optimality tolerance for the MILP solver, ϵ , which is particularly useful in the case of integer data for the rectangles dimensions like in the problems considered in this article, i.e. $\epsilon=0.999$. The detailed search algorithm for the continuous-space model NCS is given in Figure 8.

$$H \leq \text{cutoff} \tag{31}$$

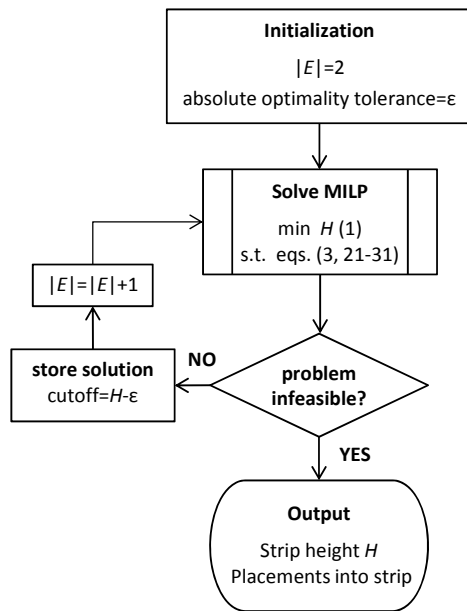


Figure 8. Search algorithm for new slot-based continuous-space approach (NCS).

6.2. The Temporary Plateau Limitation

Although the iterative search procedure works quite well in practice (the first iterations can be solved rather rapidly), it is important to highlight that the plateau in the objective function may only be temporary, meaning that there is no guarantee that the global optimal solution will result even if all iterations are solved to optimality. In fact, this phenomenon was observed in 2 out of the 29 test problems solved, contrary to what happened with the hybrid discrete/continuous-space model from (Castro & Oliveira, 2011), where a simple example had to be used for illustrative purposes.

Taking the well-known `ngcut08` problem as an example, the search for the optimal strip height starts with $H=149$ for 2 allocation points. For a single increment in the number of allocation points, there is a substantial decrease in height, $H=75$. The solution continues to improve for 4 and 5 points, with the optimal arrangements given in Figure 9. Notice that the solution for $|E|=4$ still generates a considerable amount of waste and that the location of e_2 and e_3 define guillotine cuts. The same can be said for e_2 and e_3 concerning the solution on the right but not for e_4 , since rectangle I2 is located between e_3 and e_5 .

The search continues for $|E|=6$ that features an optimal strip height equal to 36. The same optimal solution is returned for $|E|=7$ leading to the termination of the search with a best solution that is in fact suboptimal. More specifically, if model NCS is solved for $|E|=8$, the global optimal solution can be found ($H=33$). These two solutions are shown in Figure 10. In the one on the right, there are already seven rectangles (I2-I4, I7, I8, I10, I12) for which the active binary variable $\bar{N}_{i,e,e'}$ features $e' > e+1$. In particular, rectangle I8 spreads across three slots, with the left-edge assigned to allocation point e_5 and the right-edge to point e_8 .

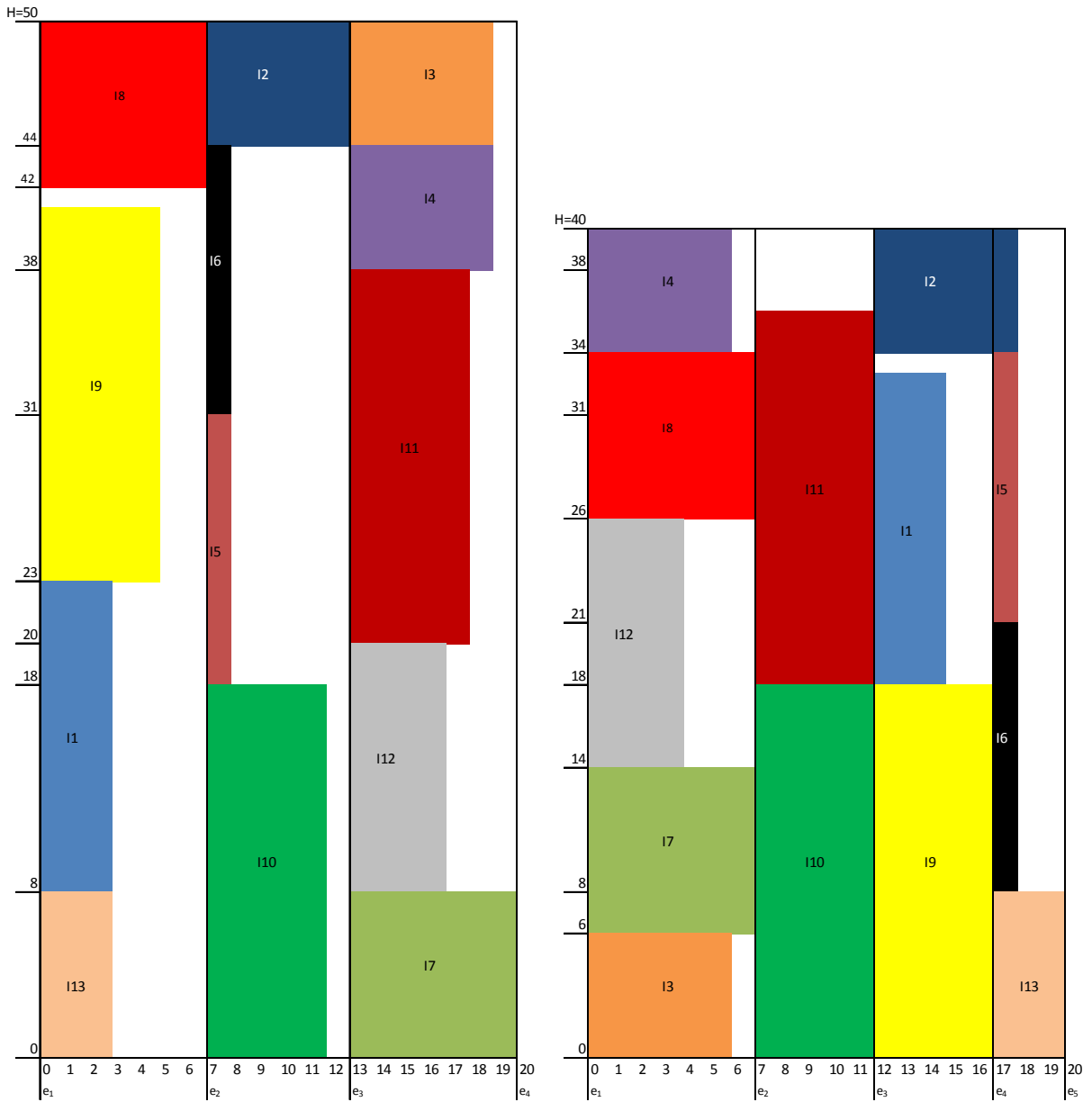


Figure 9. Influence on the number of allocation points on the optimal solution for Ex16, part 1. On the left, $|E|=4$ and $H=50$. On the right $|E|=5$ and $H=40$.

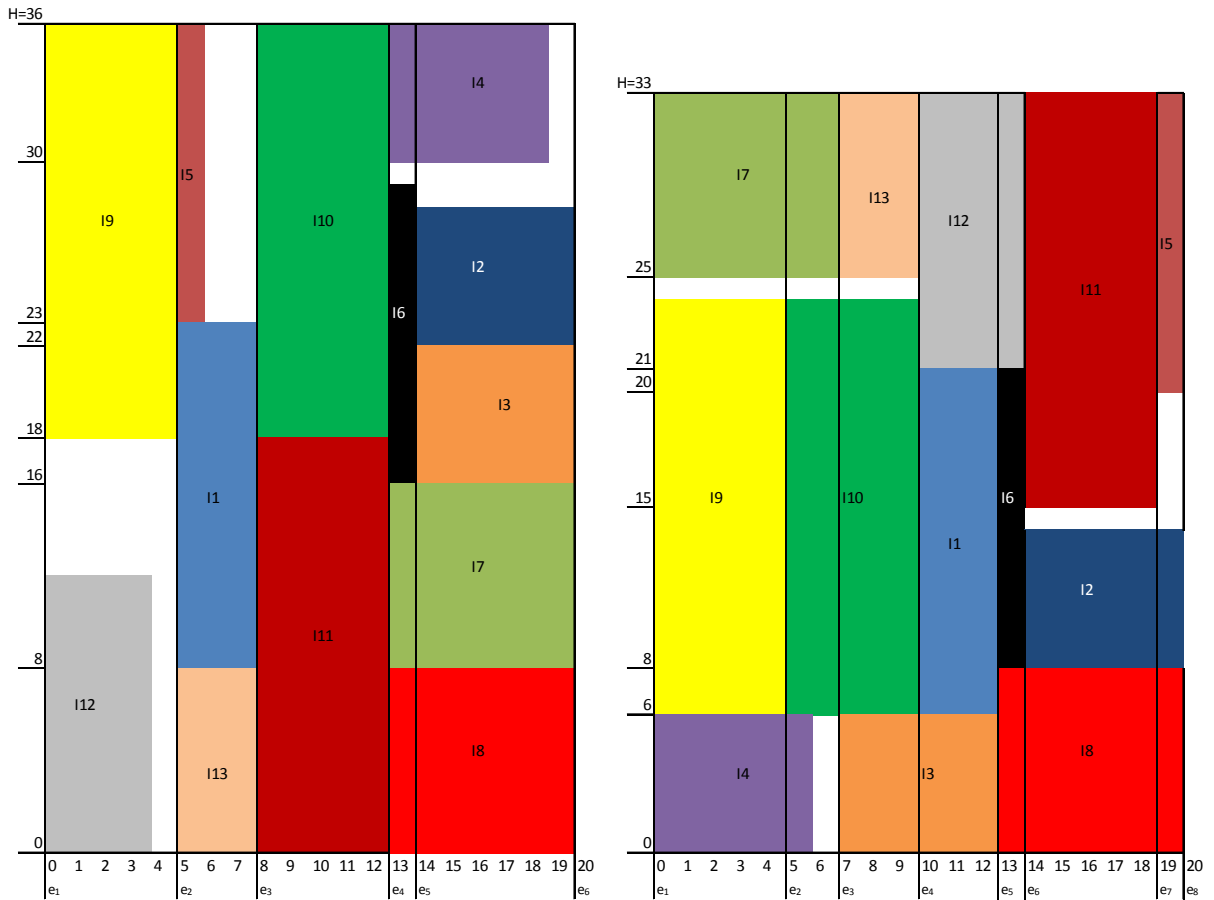


Figure 10. Influence on the number of allocation points on the optimal solution for Ex16, part 2. On the left, $|E|=6$ and $H=36$. On the right $|E|=8$ and $H=33$, which is a global optimal solution.

7. Computational Results

The performance of the given solution approaches (NDCS, NCS and CS) is now illustrated through the solution of 29 strip packing problems and compared to that of two recent solution strategies (DS and DCS, see Castro & Oliveira, 2011) that have been evaluated under the same software and hardware. More specifically, the models and search algorithms were implemented in GAMS 23.2 using CPLEX 12.1 as the MILP solver with a single thread, an absolute optimality tolerance equal to 0.999 (recall that all data are integer) and a maximum computational time per run equal to 7200 CPU s. In the case of the search algorithms, a new iteration started only if the accumulated computational time was below 7200 CPU s, making it possible for the total computational time to go up to 4 hours. In eqs. (4-5, 14-15, 26-27) we have used $M=300$ except for *cgcut02* ($M=3000$ so that the solution for $|E|=2$ is feasible). The hardware consisted of a laptop with

an Intel Core2 Duo T9300 2.5 GHz processor with 4 GB of RAM running Windows Vista Enterprise.

The GAMS input files of all five mathematical programming models and associated search algorithms are available through the CMU-IBM CyberInfrastructure for MINLP collaborative site www.minlp.org in the MINLP library of problems under the title: 2-D Orthogonal Strip Packing.

7.1. Key performance Indicators

The ultimate goal of any optimization approach is to determine the global optimal solution, which first involves finding such solution and then proving it cannot be improved further. For two of the approaches under evaluation (CS and NDCS) this involves solving a simple MILP problem per example, while for the others a few iterations may be involved. In particular, it is possible for DCS and NCS to terminate with a suboptimal solution since the search strategy for the global optimal solution is not rigorous but heuristic. In fact, *ngcut08* and *ht02* led to temporary plateaus for NCS (see section 6.2), while the phenomena was not observed for DCS. When failing to find the optimal solution, most methods are capable of finding near optimal solutions, the exception being DS for which the first feasible solution is also optimal. Table 2 provides the values for a comparative analysis of these 4 performance indicators.

Table 2. Key performance indicators on the solution of 29 test problems

Approach	Reference	Optimal solutions		Suboptimal solutions	No solutions	Best performer
		Found	Proven			
Discrete-space (DS)	(Castro & Oliveira, 2011)	19	19	-	10	8
Hybrid discrete/continuous (DCS)	(Castro & Oliveira, 2011)	21	13	8	-	7
Continuous-space (CS)	(Castillo et al. 2005)	16	6	13	-	5
New hybrid discrete/continuous (NDCS)	This work	21	19	8	-	7
New continuous-space (NCS)	This work	17	15	12	-	2
					Total	29

Table 3. Best solution found and total computational effort for different approaches (best performer in bold, suboptimal solutions in italic)

Approach	Previous work			New work		Previous work			New work			
	DS	DCS	CS	NDCS	NCS	DS	DCS	CS	NDCS	NCS		
Problem	$ I $	W	H	CPU s								
Ex1	9	18	18	18	18	18	18	4.86	3.21	0.39	2.10	4.23
Ex2	9	18	18	18	18	18	18	3.45	2.99	0.39	0.75	4.31
Ex3	10	20	23	23	23	23	23	6.21	9578 ^a	7200 ^a	244	117
Ex4	11	20	-	22	22	22	22	7205 ^a	10762 ^a	7200 ^a	340	3230
Ex5	21	10	24	24	24	24	24	4.61	2.93	46.9	41.2	50.1
SCP16	14	6	33	33	33	33	33	5.30	2.87	4335 ^b	2.87	505
cgcut01	16	10	23	23	23	23	23	2.37	4.75	7200 ^a	57.2	8027 ^a
cgcut02	23	70	-	67	68	72	74	7463 ^a	7201^a	7200 ^a	7200 ^a	8720 ^a
ngcut01	10	10	23	23	23	23	23	81.6	461	293	2.71	3.46
ngcut02	17	10	-	30	30	30	30	7202 ^a	12368 ^a	5182 ^b	26.2	329
ngcut03	21	10	28	28	29	29	29	12.6	2.87	7200 ^a	7200 ^a	13682 ^a
ngcut04	7	10	20	20	20	20	20	1.22	5.90	0.96	0.56	9.93
ngcut05	14	10	36	36	36	36	36	3.18	0.62	7200 ^a	0.81	2036
ngcut06	15	10	-	31	31	31	31	3742 ^c	8194 ^a	6235 ^b	87.5	159
ngcut07	8	20	20	20	20	20	20	0.98	3.37	0.23	0.4	1.24
ngcut08	13	20	33	33	34	33	36	29.5	7363 ^a	7200 ^a	118	1280
ngcut09	18	20	-	53	51	53	52	7210 ^a	7200 ^a	7200^a	7200 ^a	10616 ^a
ngcut10	13	30	-	80	80	80	80	8760 ^a	7200 ^a	4198 ^c	5118 ^b	3665
ngcut11	15	30	-	52	52	52	54	7215 ^a	10028 ^a	7200 ^a	7200^a	12153 ^a
ngcut12	22	30	-	87	87	87	87	10339 ^a	7202 ^a	5235 ^b	987	449
ht01	16	20	20	20	21	20	20	2.08	2.01	7200 ^a	26.6	9128 ^a
ht02	17	20	20	21	21	20	21	5.28	7200 ^a	7200 ^a	6952	4093
ht03	16	20	20	20	21	20	20	2.51	9.34	7200 ^a	20.4	1226
ht04	25	40	15	16	17	17	17	91.8	7202 ^a	7200 ^a	7200 ^a	7203 ^a
ht05	25	40	15	16	17	17	17	20.4	7200 ^a	7200 ^a	7200 ^a	12737 ^a
ht06	25	40	15	15	16	15	16	18.3	11.5	7200 ^a	773	13428 ^a
ht07	28	60	30	31	34	38	35	3771	7201 ^a	7200 ^a	7200 ^a	7203 ^a
ht08	29	60	-	31	33	35	37	7200 ^a	7201^a	7200 ^a	7204 ^a	7200 ^a
ht09	28	60	-	33	32	39	34	7200 ^a	7201 ^a	7200^a	7204 ^a	12752 ^a

^a Maximum computational time termination. ^b Out of memory termination. ^c Abnormal termination with suboptimal solution.

The new hybrid discrete/continuous-space model (NDCS) is the best overall performer. It is able to find the same 21 optimal solutions as its DCS counterpart and was able to prove optimality in 6 more cases. When added to the fact that no iterative procedure for the global optimal solution is needed, it shows that it is preferable to use sequencing variables than multiple spatial grids for the y-axis. The same number of proven optimal solutions (19) was obtained by the discrete-space approach, whose main drawback is either finding an optimum or no solution at all. It thus has to be considered the third best performer. Further down the line come the two continuous-space approaches that exhibited a lower success rate in terms of finding and proving optimality, with the

new one having the edge due to the higher number of proven optimal solutions (15 vs. 6). It is thus better to employ a single time grid on the x -axis than positioning variables despite the disadvantage of the heuristic search procedure.

The last column in Table 2 gives the number of problems in which a given formulation was the best performer. To make the decision, we first considered solution quality and then computational time (see Table 3). Whenever two approaches had the same time (e.g. DCS and NDCS in SCP16) we chose the one not requiring an iterative search procedure (NDCS). In *ngcut11*, we broke the tie between CS and NDCS by picking the one with the highest relaxation (best possible solution) at the time of termination (43.0 vs. 50.8).

The most interesting result is that a particular formulation is the best performer in at least two problems meaning that they all can be potentially useful. In the days of parallel computing where even a relative inexpensive computer features a few threads, it is already possible to rely on multiple approaches for decision-making. Nevertheless, a few recommendations can be made that are the subject of section 8. But first, let us discuss the computational statistics related to problem size.

7.2. Computational Statistics

In Table 4, we show the number of entities related to problem size together with the root node relaxation for the models that do not require an iterative search procedure, CS and NDCS (note that the discrete-space model DS uses a different objective function and that the relaxation for the slot based models DCS and NCS changes with $|E|$). The values in columns 2 and 3 show that the new hybrid discrete/continuous-space MILP model is considerably tighter than the continuous-space model of Castillo et al. (2005). In fact, the relaxation of the former is equal to the well-known continuous lower bound (Martello et al. 2003) given by eq. (32). The same applies to hybrid model DCS, while the relaxation of the new continuous-space approach (NCS) lies somewhere in between CS and NDCS. As an example, the largest integrality gap was for *ngcut12*, with an optimal solution $H=87$ and a root node relaxation equal to 24 (CS), 41.6 (NCS) and 83.5 (DCS and NDCS).

$$\sum_{i \in I} w_i h_i / W \tag{32}$$

With respect to problem size, the option of discretizing both dimensions leads to the largest number of binary and total variables, which can easily go past the tens of thousands listed in Table 4. This was hardly unexpected considering the use of three-index binary variables (i,x,y) in DS, where $|X|=W$ and $|Y|=H$. DCS also features three-index binaries (i,x,e) but now the number of allocation points in the y -axis required to find the optimal solution is considerably lower than the strip height. In terms of the models given in this paper, NCS also features three indices (i,e,e') but two of them are event based, resulting in typically fewer binary variables than the two-index (i,x) model NCS. The lowest number of binaries often results from the two-index (i,i') model CS.

A similar trend is observed in terms of total variables while for total constraints the results are not as conclusive with DS, DCS and CS sharing top spot. The number of constraints for the new hybrid model (NDCS) is typically one order of magnitude larger, with NCS lying in fourth place. Notice that the large majority of the constraints for NDCS arise from the y -axis no overlap constraints, which feature three-indices (x,i,i') , whereas in the corresponding NCS constraints, the first is an allocation point index (e,i,i') . The other three approaches have just two indices in the no overlap constraints: one set of (x,y) equations in DS; 3 sets of (x,e) constraints in DCS; and 2 sets of (i,i') equations in CS. With respect to the comparison between the two hybrid discrete/continuous-space models, moving from DCS to NDCS can be viewed as switching the complexity from the binary variables to the constraints side. Overall, it is clear that an analysis based solely on problem size is hardly suitable to predict the best performer.

Table 4. Statistics for last problem solved (RMIP=root node relaxation, DV=discrete variables, TV=total variables, SE=single equations)

Approach	CS	NDCS	DS	DCS	CS	NDCS	NCS	DS	DCS	CS	NDCS	NCS	DS	DCS	CS	NDCS	NCS
Problem	RMIP		DV					TV					SE				
Ex1	15	15.4	1342	580	72	152	225	1667	671	91	162	241	334	359	180	1332	645
Ex2	15	15.4	1323	585	72	153	225	1648	676	91	163	241	334	359	180	1332	645
Ex3	11	20	2485	652	90	208	255	2946	733	111	219	272	471	322	220	1840	776
Ex4	13	20	2428	720	110	235	363	2849	801	133	247	382	432	323	264	2242	1107
Ex5	9	22.5	3729	1336	420	377	525	3970	1417	463	399	552	262	325	924	4252	2462
SCP16	11	31.8	1829	441	182	154	301	2028	484	211	169	321	213	168	420	1126	1153
cgcut01	8	22.5	2409	868	240	244	456	2640	939	273	261	479	247	282	544	2442	1814
cgcut02	31	62.1	63438	6425	506	1538	598	67849	6776	553	1562	627	4434	1413	1104	35536	2926
ngcut01	10	20	1069	420	90	105	145	1300	491	111	116	160	241	276	220	930	484
ngcut02	9	29	2582	1080	272	244	391	2873	1181	307	262	414	308	397	612	2764	1654
ngcut03	9	27.7	3820	1296	420	372	651	4101	1377	463	394	679	302	325	924	4252	3009
ngcut04	15	16.2	606	186	42	83	168	807	217	57	91	182	208	121	112	444	419
ngcut05	12	35.3	2954	560	182	205	385	3315	621	211	220	406	375	204	420	1858	1420
ngcut06	11	29	2391	792	210	204	330	2692	873	241	220	351	316	319	480	2140	1310
ngcut07	20	20	1603	393	56	159	148	2004	454	73	168	162	409	242	144	1156	421
ngcut08	18	31.7	4780	856	156	292	351	5441	937	183	306	371	674	325	364	3166	1241
ngcut09	20	48.7	10428	1578	306	416	532	11409	1699	343	435	556	999	486	684	6176	2256
ngcut10	30	57.5	12285	1512	156	330	351	14536	1693	183	344	371	2264	721	364	4736	1241
ngcut11	29	49.4	11076	1920	210	425	525	12577	2101	241	441	548	1516	723	480	6360	1927
ngcut12	24	83.5	23813	4277	462	560	561	26334	4668	507	583	589	2543	1556	1012	13934	2689
ht01	12	20	3907	1305	240	381	568	4308	1406	273	398	592	417	406	544	4852	2167
ht02	13	20	4222	1305	272	397	493	4623	1406	307	415	517	418	407	612	5494	2029
ht03	14	20	4054	1488	204	368	456	4455	1609	273	385	479	417	484	544	4852	1814
ht04	5	15	10691	4140	600	1128	675	11292	4341	651	1154	706	626	815	1300	24090	3430
ht05	7	15	10938	4230	600	1146	825	11539	4431	651	1172	857	626	815	1300	24090	4181
ht06	7	15	11131	5010	600	1135	825	11732	5251	651	1161	857	626	973	1300	24090	4181
ht07	13	30	35531	8394	756	1777	798	37332	8755	813	1806	832	1829	1456	1624	45476	4261
ht08	11	30	37249	7400	812	1886	841	39050	7701	871	1916	876	1830	1219	1740	48838	4558
ht09	14	30	35415	7055	756	1789	966	37216	7356	813	1819	1001	1829	1218	1624	45476	5186

Table 5. Main Characteristics of Tested Approaches for 2-D Strip Packing Problems

Spatial Approach	Full Discrete	Hybrid discrete/continuous		Full Continuous	
Name	DS	DCS	NDCS	CS	NCS
x-axis	Single uniform grid	Single uniform grid	Single uniform grid	General precedence	Single slot-based grid
y-axis	Single uniform grid	Multiple slot-based grid	General precedence	General precedence	General precedence
A priori decisions that can affect solution	Size of squares (rectangles) in the mesh	Width of uniform time slots (x-axis); number of allocation points of different grids (y-axis)	Width of uniform time	None	Number of allocation points and number of slots a rectangle can span (x-axis)
Need for iterative search procedure	Yes	Yes	No	No	Yes
Strengths	Perfect packing problems; possibility of using a courser grid to keep problem tractable (data rounded to multiples of slot width and height).	Tightest formulation (same relaxation as NDCS)	Best overall performer; tightest formulation (same relaxation as DCS)	Ability to find good solutions fast; leads to the smallest problem sizes	Problem size independent of strip dimensions.
Drawbacks	Optimal solution is the first feasible solution from search procedure; can lead to prohibitively large problem sizes when considering large strip areas and accurate data; solution dependence on mesh size	Solution dependence on number of allocation points; Strip width affects problem size	Strip width affects problem size	Not particularly tight meaning that global optimality may be difficult to prove	Iterative search procedure may end with suboptimal solution

8. Overview of the Main Features of Alternative Approaches

The characteristics, strengths and drawbacks of the five alternative approaches tested are summarized in Table 5. The first aspect to highlight is related to the motivation behind this paper and concerns the type of spatial representation employed. The ideal formulation should be fully continuous-space to be independent of problem data and allow considering real and not just integer data for the rectangles widths and heights. If not, we may need to employ the finest discretization possible (e.g. strip divided in 1×1 squares) to consider the real accurate problem, which may lead to grids with too many slots, in other words, to an intractable problem. This property can also be considered an advantage since it gives an obvious way to reduce the problem size and hence complexity. By increasing the slot size (coarser grid) and rounding up problem data, an approximation version of the problem can still be solved. While it is beyond the scope of this paper to evaluate this option, it is relevant to highlight that this strategy is frequently employed for the effective solution of industrial sized scheduling problems (Méndez et al. 2006).

A full discrete-space approach (DS) has been shown particularly efficient in zero-waste, perfect packing problems, where adding the constraint of no empty elementary squares has had a major impact on efficiency (Castro & Oliveira, 2011). Discrete approaches are known to be considerably tighter than their continuous counterparts thus compensating the larger size of their resulting mathematical programming problems. By keeping one of the dimensions discrete (x -axis), we have kept the relaxation as tight as possible, equal to the continuous lower bound. This is somewhat related to a higher likelihood of finding and proving optimality as can be seen in Table 2. The advantage of using general precedence sequencing variables instead of multiple grids in the y -axis is that it avoids the heuristic search procedure for the global optimal solution, which can in theory lead to termination with a suboptimal solution (Castro & Oliveira, 2011).

Such a drawback was indeed observed for the new continuous-space formulation (NCS), suggesting that it may be more frequent when using single rather than multiple grids. Nevertheless, NCS was tighter than its general precedence counterpart (CS), which can explain why more problems were solved to global optimality. However, both are considerably less tight than DCS and

NDCS. The worst performance of CS in terms of proving optimality is compensated by the generation of the smallest MILPs of the lot, which is translated into the ability of finding very good solutions in the early nodes of the search tree.

9. Conclusions

This paper has presented two new mixed-integer linear programming approaches for the 2-dimensional strip packing problem, the NDCS and NCS models. Contrary to the approach of relying on two sets of binary positioning variables to locate a particular rectangle with respect to another, we keep the set of positioning variables for the y -axis but use a different approach for the x -axis. While in one model (NDCS) the x -axis is discretized through the use of a uniform spatial grid consisting of a few slots, in the other (NCS), a non-uniform continuous time grid with fewer slots is employed. Both can be viewed as hybrid models in the sense that different concepts for spatial representation are being combined.

The performance of the new models has been tested in several problems taken from the literature and compared to three other MILP-based approaches both quantitatively, by using the same hardware and software, and qualitatively, by highlighting the main advantages and limitations of each particular approach. The new hybrid discrete/continuous-space model was shown to be the best performer in key performance indicators like number of optimal solutions found and proven, while the new continuous-space approach more than doubled the number of problems solved to optimality with respect to a previously published continuous-space model with two sets of positioning variables. Overall, it was interesting to find out that all tested methods can be potentially useful since each approach was the best performer in at least 2 out of 29 problems.

Acknowledgments

The authors gratefully acknowledge financial support from Luso-American Foundation and the National Science Foundation under the 2011 Portugal – U.S. Research Networks Program.

References

- Alvarez-Valdes, R., Parreño, F., & Tamarit, J.M. (2009). A Branch and Bound Algorithm for the Strip Packing Problem. *OR Spectrum*, 31, 431-459.
- Amossen, R.R., & Pisinger, D. (2010). Multi-dimensional Bin Packing Problems with Guillotine Constraints. *Computers & Operations Research*, 37, 1999-2006.
- Bekrar, A., Kacem, I., & Chu, C. (2007). A Comparative Study of Exact Algorithms for the Two Dimensional Strip Packing Problem. *Journal of Industrial and Systems Engineering*, Vol. 1, No. 2, 151-170.
- Capón-Garcia, E., Ferrer-Nadal, S., Graells, M., Puigjaner L. (2009). An Extended Formulation for the Flexible Short-Term Scheduling of Multiproduct Semicontinuous Plants. *Ind. Eng. Chem. Res.* 48, 2009-2019.
- Capón-Garcia, E., Moreno-Benito, M., Espuña A. (2011). Improved Short-Term Batch Scheduling Flexibility using Variable Recipes. *Ind. Eng. Chem. Res.* 50, 4893-4992.
- Castillo, I., Westerlund, J., Emet, S., Westerlund, T. (2005). Optimization of block layout design problems with unequal areas: A comparison of MILP and MINLP optimization models. *Comput. Chem. Eng.*, 30, 54-69.
- Castro, P., Barbosa-Póvoa, A.P.F.D, & Matos, H. (2001). An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants. *Industrial and Engineering Chemistry Research*, 40, 2059 – 2068.
- Castro, P.M., Barbosa-Póvoa, A.P., Matos, H.A., & Novais, A.Q. (2004). Simple Continuous-Time Formulation for Short-Term Scheduling of Batch and Continuous Processes. *Ind. Eng. Chem. Res.*, 43, 105-118.
- Castro P.M., & Oliveira, J.F. (2011). Scheduling Inspired Models for Two-Dimensional Packing Problems. *European Journal of Operational Research*, 215, 45-56.
- Castro, P.M., & Novais, A.Q. (2008). Short-Term Scheduling of Multistage Batch Plants with Unlimited Intermediate Storage. *Ind. Eng. Chem. Res.*, 47, 6126-6139.
- Castro, P.M., Novais, A.Q., & Carvalho, A. (2008). Optimal Equipment Allocation for High Plant Flexibility: An Industrial Case Study. *Ind. Eng. Chem. Res.* 47, 2742-2761.
- Castro, P.M., Harjunkoski, I., & Grossmann, I.E. (2009). New Continuous-Time Scheduling Formulation for Continuous Plants under Variable Electricity Cost. *Ind. Eng. Chem. Res.* 48: 6701-6714.

- Castro, P.M., Harjunkoski, I., & Grossmann, I.E. (2011). Optimal Scheduling with Energy Constraints. *Computers & Chemical Engineering*, 35, 372-387.
- Dimitriadis, A.D., Shah, N., & Pantelides, C.C. (1997). RTN-based Rolling Horizon Algorithms for Medium Term Scheduling of Multipurpose Plants. *Computers & Chemical Engineering*, 21, S1061.
- Ferrer-Nadal, S., Capón-García, E., Méndez, C.A., & Puigjaner, L. (2008). Material Transfer Operations in Batch Scheduling. A Critical Modeling Issue. *Ind. Eng. Chem. Res.* 47, 7721-7732.
- Giannelos, N. F., & Georgiadis, M. C. (2002). A Simple Continuous-Time Formulation for Short-Term Scheduling of Multipurpose Batch Processes. *Ind. Eng. Chem. Res.* 41, 2178.
- Giménez, D.; Henning, G.; Maravelias, C. (2009). A Novel Network-based Continuous-Time Representation for Process Scheduling: Part I. Main Concepts and Mathematical Formulation. *Computers and Chemical Engineering*, 33, 1511-1528.
- Glismann, K., Gruhn, G. (2001). Short-term Scheduling and Recipe Optimization of Blending Processes. *Computers and Chemical Engineering* 25, 627-634.
- Grancolas, S., & Pinto, C. (2010). A SAT Encoding for Multi-dimensional Packing Problems. In: Lodi A, Milano M, Toth P, editors. *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, Berlin Heidelberg: Springer-Verlag, 141-146.
- Gupta, S., & Karimi, I. A. (2003). An improved MILP formulation for scheduling multiproduct, multistage batch plants. *Industrial and Engineering Chemistry Research*, 42, 2365–2380.
- Harjunkoski, I., & Grossmann, I. E. (2002). Decomposition Techniques for Multistage Scheduling Problems using Mixed-integer and Constraint Programming Methods. *Comput. Chem. Eng.* 26, 1533-1552.
- Ierapetritou, M.G., & Floudas, C.A. (1998). Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes. *Industrial and Engineering Chemistry Research*, 37, 4341 – 4359.
- Kenmochi, M., Imamichi, T., Nonobe, K., Yagiura, M., & Nagamochi, H. (2009). Exact Algorithms for the Two-Dimensional Strip Packing Problem with and without Rotations. *European Journal of Operational Research*, 198, 73-83.
- Kondili, E., Pantelides, C.C., & Sargent, W. H. (1993). A general algorithm for short-term scheduling of batch operations – I. MILP formulation. *Computers and Chemical Engineering*, 2, 211-227.

Li, J., Floudas, C.A. (2010). Optimal Event Point Determination for Short-Term Scheduling of Multipurpose Batch Plants via Unit-Specific Event-Based Continuous-Time Approaches. *Ind. Eng. Chem. Res.*, 49, 7446-7469.

Liu, Y., & Karimi, I. A. (2007). Scheduling multistage, multiproduct batch plants with nonidentical parallel units and unlimited intermediate storage. *Chemical Engineering Science*, 62, 1549–1566.

Maravelias, C. T. & Sung, C. (2009). Integration of production planning and scheduling: Overview, Challenges and Opportunities. *Computers and Chemical Engineering*, 33, 1919-1930.

Maravelias, C.T., & Grossmann, I. E. (2003a). Minimization of the Makespan with a Discrete-Time State-Task Network Formulation. *Ind. Eng. Chem. Res.* 42, 6252.

Maravelias, C.T., & Grossmann, I. E. (2003b). A New General Continuous-Time State Task Network Formulation for Short Term, Scheduling of Multipurpose Batch Plants. *Ind. Eng. Chem. Res.*, 42, 3056-3074.

Martello, S., Monaci, M., & Vigo, D. (2003). An exact approach to the strip-packing problem. *Inform Journal on Computing*, 15, 310-319.

Méndez, C. A., Henning, G. P., & Cerdá, J. (2000). Optimal scheduling of batch plants satisfying multiple product orders with different due-dates. *Computers and Chemical Engineering*, 24, 2223–2245.

Méndez, C. A., Henning, G. P., & Cerdá, J. (2001). An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities. *Computers and Chemical Engineering*, 25, 701–711.

Méndez, C.A., Cerdá, J., Grossmann, I.E., Harjunkoski, I., & Fahl, M. (2006). State-of-the-art Review of Optimization Methods for Short-term Scheduling of Batch Processes. *Comput. Chem. Eng.* 30, 913-946.

Ortmann, F.G., Ntene, N., & van Vuuren, J.H. (2010). New and Improved Level Heuristics for the Rectangular Strip Packing and Variable-sized Bin Packing Problems. *European Journal of Operational Research*, 203, 306-315.

Pantelides, C.C. (1994). Unified frameworks for optimal process planning and scheduling. In: *Proceedings of the Second Conference on Foundations of Computer-Aided Process Operations*, CACHE publications, New York, 253–274.

- Pinto J. & Grossmann, I. A. (1995). Continuous Time Mixed Integer Linear Programming Model for the Short-Term Scheduling of Multistage Batch Plants. *Ind. Eng. Chem. Res.* 34, 3037.
- Prasad, P.; Maravelias, C.T. (2008). Batch selection, assignment and sequencing in multi-stage, multi-product processes. *Comput. Chem. Eng.* 32: 1114-1127.
- Sawaya, N.W., & Grossmann, I.E. (2005). A Cutting Plane Method for Solving Linear Generalized Disjunctive Programming Problems. *Comput. Chem. Eng.*, 29, 1891-1913.
- Schilling, G., & Pantelides, C.C. (1996). A Simple Continuous-Time Process Scheduling Formulation and a Novel Solution Algorithm. *Comput. Chem. Eng.* 20, S1221-S1226.
- Shah, N., Pantelides, C.C., & Sargent, R. (1993). Optimal Periodic Scheduling of Multipurpose Batch Plants. *Annals of Operations Research*, 42, 193.
- Shaik, M., & Floudas, C. (2009). Novel Unified Modeling Approach for Short-Term Scheduling. *Ind. Eng. Chem. Res.* 48, 2947.
- Sundaramoorthy, A. & Maravelias, C.T. (2011b). Computational Study of Network-Based Mixed-Integer Programming Approaches for Chemical Production Scheduling. *Ind. Eng. Chem. Res.* 50, 5023-5040.
- Sundaramoorthy, A., & Maravelias, C. T. (2011a). A General Framework for Process Scheduling. *Aiche Journal* 57(3), 695-710.
- Sundaramoorthy, A., Karimi, I.A. (2005). A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. *Chem. Eng. Sci.* 60: 2679-2702.
- Sundaramoorthy, A.; Maravelias, C.T. (2008). Simultaneous batching and scheduling in multistage multiproduct processes. *Ind. Eng. Chem. Res.* 47: 1546-1555.
- Susarla, N., Li, J., & Karimi, I.A. (2010). A Novel Approach to Scheduling Multipurpose Batch Plants Using Unit-Slots. *AIChE J.* 56, 1859-1879.
- Wascher, G., Haussner, H., & Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183 (3), 1109–1130.
- Wassick, J. (2009). Enterprise-wide optimization in an integrated chemical complex. *Computers & Chemical Engineering*, 33, 1950-1963.
- Wassick, J.M., & Ferrio, J. (2011). Extending the Resource Task Network for Industrial Applications. *Computers and Chemical Engineering* 35, 2124-2140.

Wei, L., Zhang, D., & Chen, Q. (2009). A Least wasted First Heuristic Algorithm for the Rectangular Packing Problem. *Computers & Operations Research*, 36, 1608-1614.

Westerlund, J. (2005). Aspects on N-dimensional allocation. PhD Thesis. Faculty of Chemical Engineering, Åbo Akademi University, Finland.

Westerlund, J., Papageorgiou, L.G., & Westerlund, T. (2007). A MILP model for N-dimensional allocation. *Comput. Chem. Eng.*, 31, 1702-1714.

Wu, Y., Li, W., Goh, M., & Souza, R. (2010). Three-dimensional bin packing problem with variable bin height. *European Journal of Operational Research*, 202, 347-355.

Zhang, X., & Sargent, R. W. H. (1996). The Optimal Operation of Mixed Production Facilities General Formulation and Some Approaches for the Solution. *Comput. Chem. Eng.*, 20, 897-904.