Online Supplement for The Meet-in-the-Middle Principle for Cutting and Packing Problems

Jean-François Côté
CIRRELT, Université Laval, 2325, rue de la Terrasse, Québec, Canada G1V 0A6, jean-francois.cote@fsa.ulaval.ca

Manuel Iori
University of Modena and Reggio Emilia, via Amendola 2, 42122 Reggio Emilia, Italy, manuel.iori@unimore.it

---

Supplementary Algorithms

As discussed in Section 2.2 of the paper, for a general cutting and packing packing problem the computation of the regular normal patterns $\mathcal{B}_i$, for any item i in the item set I, may be obtained by invoking Algorithm 1 as `NormalPatterns`$(I \setminus \{i\}; W - w_i)$. Then, the computation of the entire set is simply obtained by letting $\mathcal{B} = \cup_i \mathcal{B}_i$. As described in Section 4 of the paper, the input item set I of the one-dimensional *cutting stock problem* (CSP) consists of m item types, where each type i comprises d_i items having the same width w_i. For the CSP, cutting items according to a given order (the one usually adopted is by non-increasing width) preserves optimality, and consequently the set $\mathcal{B}$ may be replaced by a smaller, or at least equivalent, set $\mathcal{B}'$ that considers such order, see Equation (15) of Section 4 of the paper.

The computation of $\mathcal{B}'$ may be obtained by invoking Algorithm 4 below. A support array T is used to store all feasible item widths combinations. T is first initialized to consider only the empty bin filling. Then, in the main loop starting at step 5, T is updated by considering all item widths combinations and the set of patterns is built accordingly. Note that, for each item type $i > 1$, the set $\mathcal{B}'_i$ is computed incrementally starting from the support array T obtained during the computation of the previous set $\mathcal{B}'_{i-1}$.

In a similar way, also the computation of the minimal set of MIM patterns for the CSP may benefit from the adopted item sorting. This can be done by using Algorithm 5 below, which is an adaption to the CSP of Algorithm 3 of Section 3.2 of the paper. Algorithm 5 starts by computing the regular normal patterns at step 3. Then, it fills the support arrays for the left and right patterns, T_{left} and T_{right}, respectively, at steps 4-13. Note that the check on $w_i \leq W/2$ at step 7 is used to impose that items having width larger than $W/2$ are only packed with their lowest corner in 0 (because no larger item can be packed in the same bin and the non-increasing width order must be fullfilled). The computation of the threshold value $t_{\min}$ for which $\sum_{i \in I} |\mathcal{M}'_{is}|$ is a minimum is obtained by steps 14-21. Finally, set $\mathcal{M}'$ is built at steps 22-30.

Algorithm 4 `BPatternsCSP`$(I;\, W)$

Require: I: set of *sorted* items, W: bin width
$\mathcal{B}' \leftarrow \emptyset$
$T \leftarrow [0 \text{ to } W]$: an array with all entries initialized at 0
$T[0] \leftarrow 1$
for $i = 1$ to m **do**
    $\mathcal{B}'_i \leftarrow \emptyset$
    for $p = W - w_i$ to 0 **do**
        if $T[p] = 1$ **then**
            for $k = 1$ to d_i **do**
                if $p + w_i * k > W$ **then break**
                $T[p + w_i * k] \leftarrow 1$
                $\mathcal{B}'_i \leftarrow \mathcal{B}'_i \cup \{p + w_i * (k-1)\}$
            end for
        end if
    end for
    $\mathcal{B}' \leftarrow \mathcal{B}' \cup \mathcal{B}'_i$
end for
return $\mathcal{B}'$

Algorithm 5 `MinimalMIMSetCSP`$(I;\ W)$

Require: I: set of *sorted* items, W: bin width
$T_{left},\ T_{right} \leftarrow [0 \text{ to } W]$: two arrays with all entries initialized at zero
$\mathcal{B}' \leftarrow$ `BPatternsCSP`$(I;\ W)$
for $i = 1$ to m **do**
    for $p \in \mathcal{B}'_i$ **do**
        $T_{left}[p] \leftarrow T_{left}[p] + 1$
        if $w_i \leq W/2$ **then** $T_{right}[W - w_i - p] \leftarrow T_{right}[W - w_i - p] + 1$
    end for
end for
for $p = 1$ **to** W **do**
    $T_{left}[p] \leftarrow T_{left}[p] + T_{left}[p-1]$
    $T_{right}[W-p] \leftarrow T_{right}[W-p] + T_{right}[W-(p-1)]$
end for
$t_{\min} \leftarrow 1$
$\min \leftarrow T_{left}[0] + T_{right}[1]$
for $p = 2$ **to** W **do**
    if $T_{left}[p-1] + T_{right}[p] < \min$ **then**
        $\min \leftarrow T_{left}[p-1] + T_{right}[p]$
        $t_{\min} \leftarrow p$
    end if
end for
$\mathcal{M}' \leftarrow \emptyset$
for $i = 1$ to m **do**
    $\mathcal{M}'_i \leftarrow \emptyset$
    for $p \in \mathcal{B}'_i$ **do**
        if $p < t_{\min}$ **then** $\mathcal{M}'_i \leftarrow \mathcal{M}'_i \cup \{p\}$
        if $(W - w_i - p \geq t_{\min}$ **and** $w_i \leq W/2)$ **then** $\mathcal{M}'_i \leftarrow \mathcal{M}'_i \cup \{W - p - w_i\}$
    end for
    $\mathcal{M}' \leftarrow \mathcal{M}' \cup \mathcal{M}'_i$
end for
return $\mathcal{M}'$

Instances of the Non-Exact Two-Stage Cutting Stock Problem

Attached to this supplement, we provide the two benchmark sets of instances that were used for the computational evaluation of the MIM patters in the context of the non-exact two-stage cutting stock problem (Section 5 of the paper). Set A was used in Macedo et al. (2010), Silva et al. (2010), and Mrad et al. (2013). Set ATP originates from Alvarez-Valdes et al. (2002) and was later used in Silva et al. (2010). The format is the same for the two sets, and each instance can be read in the following way:

- name of the instance
- width and height of the bin
- number of items
- for each item:
  — width, height, and demand of the item.

References

Alvarez-Valdes R, Parajón A, Tamarit J (2002) A tabu search algorithm for large-scale guillotine (un)constrained two-dimensional cutting problems. *Computers & Operations Research* 29(7):925–947.

Macedo R, Alves C, Valério de Carvalho J (2010) Arc-flow model for the two-dimensional guillotine cutting stock problem. *Computers & Operations Research* 37(6):991–1001.

Mrad M, Meftahi I, Haouari M (2013) A branch-and-price algorithm for the two-stage guillotine cutting stock problem. *Journal of the Operational Research Society* 64(5):629–637.

Silva E, Alvelos F, Valério de Carvalho J (2010) An integer programming model for two- and three-stage two-dimensional cutting stock problems. *European Journal of Operational Research* 205(3):699–708.