1 A bi-dimensional level packing problem arising from the optimization of thermal treatment of heat exchangers

Industrial heat exchangers are huge metallic cylinders subject to high reliability requirements. One of the critical phases in their production is a thermal treatment that implies to cover the cylinder with rectangular patches. We investigate the resulting optimization problem, where the objective is to use an optimal subset of patches taken from a ground set of available patches, while complying with some geometrical constraints. The resulting mathematical model is a variation of the level rectangle packing problem, an NP-hard generalization of the bin packing problem.

2 Problem description

The surface of the heat exchanger to be covered is represented by a rectangle of known size: we indicate its width by W and its height by H. A set $\mathcal{N}$ of N types of rectangular patches is available; each patch type $i \in \mathcal{N}$ is characterized by its unit cost c_i, its width w_i and its height h_i and by a number M_i of available patches. For operational reasons it is required that patches are arranged into adjacent horizontal stripes (levels); we indicate the set of levels by $\mathcal{L}$. Patches cannot overlap. A maximum horizontal gap equal to δ^W is allowed between patches in the same level. A maximum vertical gap equal to δ_H is allowed between patches in adjacent levels.

Variables. A binary variable t_{ij} indicates whether patches of type $i \in \mathcal{N}$ are used in level $j \in \mathcal{L}$. An integer non-negative variable x_{ij} indicates how many patches of type $i \in \mathcal{N}$ are arranged into level $j \in \mathcal{L}$. The height of each level $j \in \mathcal{L}$ is indicated by a continuous non-negative variable $z_j \geq 0$. Since the number of levels is not a datum, we also use a binary variable y_j to indicate whether each level $j \in \mathcal{L}$ is actually used or not.

Constraints. Variables x and t are related by:

$$x_{ij} <= M_i t_{ij} \quad \forall i \in \mathcal{N}, j \in \mathcal{L}.$$

The number of patches for each type is limited:

$$\sum_{j \in \mathcal{L}} x_{ij} \leq M_i \quad \forall i \in \mathcal{N}.$$

The sum of the heights of the levels must be equal to the height of the rectangle:

$$\sum_{j \in \mathcal{L}} z_j = H.$$

Obviously, levels that are not used have zero height:

$$z_j \leq H y_j \quad \forall j \in \mathcal{L}.$$

Patches cannot exceed the height of their level:

$$h_i t_{ij} \leq z_j \quad \forall i \in \mathcal{N} \quad \forall j \in \mathcal{L}.$$

Vertical tolerance:

$$z_j \leq H - (H - h_i - \delta^H) t_{ij} \quad \forall j \in \mathcal{L}.$$

Horizontal tolerance:

$$\sum_{i \in \mathcal{N}} w_i x_{ij} \leq W y_j \leq \sum_{i \in \mathcal{N}} (w_i + \delta^W) x_{ij} \quad \forall j \in \mathcal{L}.$$

With this formulation of the last constraints we allow for a maximum gap of $\frac{\delta^W}{2}$ between the patches and the edge of the rectangle.

Objective. The objective is to minimize the cost of the patches used:

$$\text{minimize } f = \sum_{i \in \mathcal{N}, j \in \mathcal{L}} c_i x_{ij}.$$

Patches rotation. An important variation of the problem allows for rotating the patches. We deal with this possibility by defining two virtual patch types for each real patch type. Now $\mathcal{N}$ indicates the set of virtual patch types and all virtual patch types are given in pairs; $\mathcal{P}$ indicates the set of pairs. For each pair $p = [u, v] \in \mathcal{P}$ with $u \in \mathcal{N}$ and $v \in \mathcal{N}$, we have $w_u = h_v$, $w_v = h_u$, $c_u = c_v$, $M_u = M_v$. The only constraint that needs to be reformulated is the assignment constraint:

$$\sum_{j \in \mathcal{L}} (x_{uj} + x_{vj}) \leq M_p \quad \forall p = [u, v] \in \mathcal{P}$$

where $M_p = M_u = M_v$ is the overall number of patches of type $p \in \mathcal{P}$, and $u \in \mathcal{N}$ and $v \in \mathcal{N}$ are two virtual types corresponding to the same real type $p \in \mathcal{P}$.

Mathematical model. The following integer linear programming model is obtained.

$$\begin{aligned}
\text{minimize } f = & \sum_{i \in \mathcal{N}, j \in \mathcal{L}} c_i x_{ij} && && (1)\\
\text{s.t. } & x_{ij} <= M_i t_{ij} && \forall i \in \mathcal{N}, j \in \mathcal{L} && (2)\\
& \sum_{j \in \mathcal{L}} (x_{uj} + x_{vj}) \leq M_p && \forall p = [u, v] \in \mathcal{P} && (3)\\
& \sum_{j \in \mathcal{L}} z_j = H && && (4)\\
& z_j \leq H y_j && \forall j \in \mathcal{L} && (5)\\
& h_i t_{ij} \leq z_j && \forall i \in \mathcal{N},\ \forall j \in \mathcal{L} && (6)\\
& z_j \leq H - (H - h_i - \delta^H) t_{ij} && \forall i \in \mathcal{N},\ \forall j \in \mathcal{L} && (7)\\
& \sum_{i \in \mathcal{N}} w_i x_{ij} \leq W y_j && \forall j \in \mathcal{L} && (8)\\
& \sum_{i \in \mathcal{N}} (w_i + \delta^W) x_{ij} \geq W y_j && \forall j \in \mathcal{L} && (9)\\
& x_{ij} \in \mathcal{Z}_+ && \forall i \in \mathcal{N},\ \forall j \in \mathcal{L} && (10)\\
& t_{ij} \in \{0, 1\} && \forall i \in \mathcal{N},\ \forall j \in \mathcal{L} && (11)\\
& y_j \in \{0, 1\} && \forall j \in \mathcal{L} && (12)\\
& z_j \geq 0 && \forall j \in \mathcal{L}. && (13)
\end{aligned}$$

3 A reformulation

We present a reformulation of the problem, where each column $k \in \mathcal{K}$ corresponds to a feasible level filled with patches.

$$\begin{aligned}
\text{minimize } f = & \sum_{k \in \mathcal{K}} b_k \lambda_k && && (14)\\
\text{s.t. } & \sum_{k \in \mathcal{K}} \alpha_{pk} \lambda_k \leq M_p && \forall p \in \mathcal{P} && (15)\\
& \sum_{k \in \mathcal{K}} h_k^{min} \lambda_k \leq H && && (16)\\
& \sum_{k \in \mathcal{K}} h_k^{max} \lambda_k \geq H && && (17)\\
& \lambda_k \in \mathcal{Z}_+ && \forall k \in \mathcal{K}. && (18)
\end{aligned}$$

Here b_k represents the total cost of all patches in level k; α_{pk} is the number of patches of real type $p \in \mathcal{P}$ in level k; h_k^{min} and h_k^{max} are the minimum and maximum feasible height for level k; λ_k are integer variables, representing the number of times level k is used in the solution. The set $\mathcal{K}$ has a combinatorial size because it includes all feasible levels (index k has been dropped for convenience):

$$
\begin{aligned}
\mathcal{K} = \{ & (\alpha, h^{min}, h^{max}) \in \mathcal{Z}_+^P \times \Re_+ \times \Re_+ : \\
& a_i <= M_i d_i & \forall i \in \mathcal{N} \qquad (19) \\
& a_u + a_v = \alpha_p & \forall [u,v] \in \mathcal{P} \qquad (20) \\
& \alpha_p \leq M_p & \forall p \in \mathcal{P} \qquad (21) \\
& h_i d_i \leq h^{min} & \forall i \in \mathcal{N} \qquad (22) \\
& h^{max} \leq H - (H - h_i - \delta^H) d_i & \forall i \in \mathcal{N} \qquad (23) \\
& \sum_{i \in \mathcal{N}} w_i a_i \leq W & (24) \\
& \sum_{i \in \mathcal{N}} (w_i + \delta^W) a_i \geq W & (25) \\
& a_i \in \mathcal{Z}_+^N & (26) \\
& d_i \in \{0,1\}^N & (27) \\
\}. &
\end{aligned}
$$

The cost of a level is given by

$$b = \sum_{i \in \mathcal{N}} c_i a_i.$$

The master problem (14)-(18) is an integer knapsack problem, with some additional features. Its linear relaxation can be solved with column generation by iteratively inserting columns in a restricted linear master problem. The reduced cost of each column is given by

$$r = \sum_{i \in \mathcal{N}} c_i a_i - \sum_{p \in \mathcal{P}} \beta_p \alpha_p - \mu^{min} h^{min} - \mu^{max} h^{max},$$

where β_p is the non-positive dual variable corresponding to each constraint (15), μ_{min} is the non-positive dual variable corresponding to constraint (16) and μ_{max} is the non-negative dual variable corresponding to constraint (17).

It should be noted that constraints (16) are not likely to be active in any optimal solution of the relaxed master problem, while constraints (17) will be active for the *break item*, i.e. the level with a fractional value of λ in the optimal solution.

The pricing problem is as follows:

$$
\begin{aligned}
\text{minimize } r = & \sum_{i \in \mathcal{N}} c_i a_i - \sum_{p \in \mathcal{P}} \beta_p \alpha_p - \mu^{min} h^{min} - \mu^{max} h^{max} \\
\text{s.t. } & a_i <= M_i d_i && \forall i \in \mathcal{N} \\
& a_u + a_v = \alpha_p && \forall [u,v] \in \mathcal{P} \\
& \alpha_p \leq M_p && \forall p \in \mathcal{P} \\
& h_i d_i \leq h^{min} && \forall i \in \mathcal{N} \\
& h^{max} \leq H - (H - h_i - \delta^H) d_i && \forall i \in \mathcal{N} \\
& \sum_{i \in \mathcal{N}} w_i a_i \leq W \\
& \sum_{i \in \mathcal{N}} (w_i + \delta^W) a_i \geq W \\
& a_i \in \mathcal{Z}_+ && \forall i \in \mathcal{N} \\
& d_i \in \{0,1\} && \forall i \in \mathcal{N} \\
& \alpha_p \in \mathcal{Z}_+ && \forall p \in \mathcal{P} \\
& h^{min} \geq 0 \\
& h^{max} \geq 0
\end{aligned}
$$

We can reformulate the model, getting rid of variables α.

$$\text{minimize } r = \sum_{i \in \mathcal{N}} (c_i - \beta_p(i)) a_i - \mu^{min} h^{min} - \mu^{max} h^{max} \tag{28}$$

$$\text{s.t. } a_i <= M_i d_i \qquad \forall i \in \mathcal{N} \tag{29}$$

$$a_u + a_v = \leq M_p \qquad \forall p = [u,v] \in \mathcal{P} \tag{30}$$

$$h_i d_i \leq h^{min} \qquad \forall i \in \mathcal{N} \tag{31}$$

$$h^{max} \leq H - (H - h_i - \delta^H) d_i \qquad \forall i \in \mathcal{N} \tag{32}$$

$$\sum_{i \in \mathcal{N}} w_i a_i \leq W \tag{33}$$

$$\sum_{i \in \mathcal{N}} (w_i + \delta^W) a_i \geq W \tag{34}$$

$$a_i \in \mathcal{Z}_+ \qquad \forall i \in \mathcal{N} \tag{35}$$

$$d_i \in \{0,1\} \qquad \forall i \in \mathcal{N} \tag{36}$$

$$h^{min} \geq 0 \tag{37}$$

$$h^{max} \geq 0 \tag{38}$$

By $p(i)$ we indicate the pair $p \in \mathcal{P}$ corresponding to type $i \in \mathcal{N}$. This pricing problem is a variation of the bounded integer knapsack problem with demand constraints and with additional penalties depending on the minimum and maximum item size (patch height).

Observation. For each given set of patches (i.e. for each feasible choice of a), the optimal values of h_{min} and h_{max} are given by $h^{min} = \max_{i \in \mathcal{N}: a_i = 1}\{h_i\}$ and $h^{max} = \delta^H + \min_{i \in \mathcal{N}: a_i = 1}\{h_i\}$.

Hence, we can solve the pricing sub-problem for each choice of the minimum and maximum patch height. In the remainder we indicate these two values by $\underline{h}$ and $\overline{h}$. For each instance of the pricing subproblem all patch types $j \in \mathcal{N}$ such that h_j does not fall in the range $[\underline{h}, \overline{h}]$ are disregarded (i.e. a_j is fixed to 0). We indicate by $\hat{N} \subseteq \mathcal{N}$ the subset of types that are compatible with the choice of $\underline{h}$ and $\overline{h}$

and we indicate the resulting pricing sub-problem as *restricted* pricing sub-problem:

$$\text{minimize } r = \sum_{i \in \hat{N}} (c_i - \beta_p(i))a_i - \mu^{min}\overline{h} - \mu^{max}(\underline{h} + \delta^H) \tag{39}$$

$$\text{s.t. } a_u + a_v =\leq M_p \qquad \forall p = [u, v] \in \mathcal{P} \tag{40}$$

$$\sum_{i \in \mathcal{N}_j} w_i a_i \leq W \tag{41}$$

$$\sum_{i \in \mathcal{N}_j} (w_i + \delta^W)a_i \geq W \tag{42}$$

$$a_i \in \mathcal{Z}_+ \qquad \forall i \in \mathcal{N} \tag{43}$$

This reformulation allows to get rid of binary variables d. The terms $-\mu^{min}\overline{h} - \mu^{max}(\underline{h} + \delta^H)$ do not affect the optimal solution since they are constant. We indicate them as $\gamma(\mu^{min}, \mu^{max}, \underline{h}, \overline{h})$. We also indicate by $\overline{c}_i = c_i - \beta_p(i)$ the reduced cost of patch type $i \in \mathcal{N}$. Hence the objective function can be rewritten as

$$\text{minimize } r = \sum_{i \in \hat{N}} \overline{c}_i a_i + \gamma(\mu^{min}, \mu^{max}, \underline{h}, \overline{h}).$$

We are left with an integer knapsack problem with items partitioned into types and a limit on the number of available items for each pair of types.

If a solution exists with a negative value of r, then a new column can be generated and inserted in the restricted master problem.