On three soft rectangle packing problems with guillotine constraints

Quoc Trung Bui
Daily-Opt Joint Stock Company
trungbui@daily-opt.com

Thibaut Vidal
Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro
vidalt@inf.puc-rio.br

Minh Hoàng Hà *
University of Engineering and Technology, Vietnam National University
minhhoang.ha@vnu.edu.vn

Abstract. We investigate how to partition a rectangular region of length $L_1$ and height $L_2$ into $n$ rectangles of given areas $(a_1, \ldots, a_n)$ using two-stage guillotine cuts, so as to minimize either (i) the sum of the perimeters, (ii) the largest perimeter, or (iii) the maximum aspect ratio of the rectangles. These problems play an important role in the ongoing Vietnamese land-allocation reform, as well as in the optimization of matrix multiplication algorithms. We show that the first problem can be solved to optimality in $O(n \log n)$, while the two others are NP-hard. We propose mixed integer programming (MIP) formulations and a binary search-based approach for solving the NP-hard problems. Experimental analyses are conducted to compare the solution approaches in terms of computational efficiency and solution quality, for different objectives.

Keywords. Soft rectangle packing, guillotine constraints, complexity analysis, integer programming.

1 Introduction

We consider a family of soft rectangle packing problems in which a rectangular region of length $L_1$ and height $L_2$ must be partitioned into $n$ rectangles of positive areas $(a_1, \ldots, a_n)$, where $\sum_{i=1}^{n} a_i = L_1 \times L_2$. The areas of the rectangles are fixed, and their position, length
and height constitute the decision variables of the problem. Three different objectives are considered: minimizing the sum of the rectangle’s perimeters, the largest perimeter, or the largest aspect ratio, leading to three problems called \textsc{Col-Peri-Sum}, \textsc{Col-Peri-Max}, and \textsc{Col-Aspect-Ratio} respectively. Finally, the layout of the rectangles is subject to strict rules. As illustrated in Figure 1, the rectangles should be delimited by \textit{two-stage} guillotine cuts: first cutting the rectangular area horizontally to produce several layers (three on the figure), and then cutting each layer vertically to obtain the rectangles (ten on the figure).

Any solution of these problems can be described as a \textit{partition} \( \{S_1, \ldots, S_m\} \) of the rectangle set \( S = \{1, \ldots, n\} \) into \( m \) layers. Since the length of each layer is fixed to \( L_1 \), the height \( w(S_k) \) of a layer \( S_k \) (and therefore of all its contained rectangles) takes value

\[
w(S_k) = \frac{\sum_{i \in S_k} a_i}{L_1},
\]

and the length of each rectangle \( i \in S_k \) is \( a_i/w(S_k) \). Based on this observation, the objective of these problems is to find \( \{S_1, \ldots, S_m\} \) so as to minimize:

\[
\text{\textsc{Col-Peri-Sum}}: \quad \Phi_1 = 2 \times \sum_{k=1}^{m} \sum_{i \in S_k} \left( w(S_k) + \frac{a_i}{w(S_k)} \right)
\]

\[
= 2 \times \sum_{k=1}^{m} (|S_k|w(S_k) + L_1)
\]

\[
\text{\textsc{Col-Peri-Max}}: \quad \Phi_2 = 2 \times \max_k \max_{i \in S_k} \left( w(S_k) + \frac{a_i}{w(S_k)} \right)
\]

\[
\text{\textsc{Col-Aspect-Ratio}}: \quad \Phi_3 = \max_k \max_{i \in S_k} \left( \frac{a_i}{w(S_k)^2} \frac{w(S_k)^2}{a_i} \right).
\]
The main contribution of this paper is to characterize the computational complexity of these problems, proposing an efficient $O(n \log n)$ algorithm for COL-PERI-SUM, and demonstrating that the two other problems are NP-hard. Moreover, we introduce mixed integer programming (MIP) formulations for the NP-hard problems. Finally, we conduct experimental analyzes to determine the limit-size of the instances which can be efficiently solved, and compare the solutions obtained with different objectives.

2 Applications and related work

**Land reform in Vietnam.** Historically, the agricultural land of Vietnam has been classified into several categories, and each household has been given one plot for each land category, such that even a small agricultural field can be distributed to many households. These division rules have been applied in most provinces of Vietnam to ensure equality among households. However, this has led to a large fragmentation of the land in most provinces of Vietnam (Sundqvist and Anderson 2006, Pham et al. 2007). In these provinces, each household owns many small and scattered plots, located in different fields. The province of Vinh Phuc is a striking example: some households have up to 47 plots, and each plot has an average area of only ten square meters (Journal of Rural Economy 2008).

The land fragmentation turned out to be detrimental in the industrialized era. First, households cannot use machines to cultivate small plots, leading to a high production cost. Second, fragmented plots are costly to visit and maintain. Third, the excessive number of tracks separating the plots causes a waste of agricultural land (Heltberg 1998, Lam 2001, General Statistical Office 2004, March and MacAulay 2006, Sundqvist and Anderson 2006). Therefore, the Vietnamese government considers land fragmentation to be “a significant barrier to achieving further productivity gains in agriculture”, and initiated a land reform to deal with the situation. This reform aims to reduce the number of land categories, to merge small plots into large fields and finally to repartition these fields into larger plots for households. It has led to successful results in some provinces, as characterized by a significant increase in rice yield attaining 25% in Quang Nam province (Sundqvist and Anderson 2006, Bui et al. 2013).

The land reform involves two critical tasks: merging small plots into larger fields, and repartitioning these fields equitably while respecting the predefined quantity of land attributed to each household. In this study, we consider the case of rectangular fields, as this is the most
common in practice. The fields should be first split by parallel edge-to-edge tracks to facilitate the use of machines, and the resulting sections should then be separated into plots, leading to the two-stage guillotine constraints discussed in the introduction of this paper.

Finally, farmers and local authorities may have distinct objectives and motivations. The local authorities aim to minimize the amount of wasted land due to the creation of tracks, a goal which is captured by the \textsc{Col-Peri-Sum} objective. In contrast, the farmers wish to have their plots as square as possible to facilitate cultivation. To obtain equitable solutions, this goal can be expressed as a worse-case optimization, leading to the \textsc{Col-Peri-Max} and \textsc{Col-Aspect-Ratio} objectives. These objectives are not strongly conflicting, but they often lead to different land allocation decisions.

Land-consolidation strategies have been implemented in various other countries, e.g., in Germany (Borgwardt et al. 2014, Brieden and Gritzmann 2004, Borgwardt et al. 2011), Turkey (Cay et al. 2006, 2010, Cay and Uyan 2013, Hakli and Uuz 2017), Japan (Arimoto 2010), Cyprus (Demetriou et al. 2013), China (Huang et al. 2011), and Brazil (Gliesch et al. 2017). However, each country, depending on its own topology, culture, and practice has converged towards a different problem setting. In particular, the two-stage guillotine-cut restrictions and the objective functions relevant to the Vietnamese case have not yet been encountered in other land-consolidation applications. Still, some related problems can be found in the operations research and computer science literature, as discussed in the following.

\textbf{Soft rectangle packing problems.} Partitioning an area into polygons of fixed shape or area is a class of problems which has been regularly studied in the operations research and computational geometry literature. Beaumont et al. (2001, 2002) defined two optimization problems that seek to partition the unit square into a number of rectangles with given areas, so as to optimize parallel matrix-multiplication algorithms in heterogeneous parallel computing platforms. The first problem aims to minimize the sum of all rectangle perimeters, whereas the second problem aims to minimize the largest perimeter. These problems are special cases of \textsc{Peri-Sum} and \textsc{Peri-Max} where the general rectangular region is a square. The authors introduced an $7/4$-approximate algorithm and an $2/\sqrt{3}$-approximate algorithm for these problems, respectively.

Later on, Nagamochi and Abe (2007) considered the general \textsc{Peri-Sum}, \textsc{Peri-Max} and \textsc{Aspect-Ratio} problems without guillotine constraints. They introduced an $O(n \log n)$-time al-
algorithm which produces a 1.25-approximate solution for PERI-SUM, a $\frac{2}{\sqrt{3}}$-approximate solution for PERI-MAX, and finds a solution with aspect ratio smaller than $\max\{R, 3, 1 + \max_{i \in \{1, \ldots, n-1\}} \frac{a_{i+1}}{a_i}\}$ for ASPECT-RATIO where $R$ denotes the aspect ratio of the original rectangular area. Fügenschuh et al. (2014) also designed an $\frac{2}{\sqrt{3}}$-approximate algorithm and a branch-and-cut algorithm for PERI-SUM.

Other close variants of PERI-SUM have been studied. Kong et al. (1987, 1988) considered the problem of decomposing a square or a rectangle into a number of rectangles of equal area so as to minimize the maximum rectangle perimeter. VLSI floorplan design and facility location applications also led to a number of related studies (Young et al. 2001, Ji et al. 2017, Paes et al. 2017). Ibaraki and Nakamura (2006) proposed a local search and mathematical programming algorithm to solve rectangular packing problems where the shapes of the rectangles are adjustable within given perimeter and area constraints.

Finally, Beaumont et al. (2002) considered COL-PERI-SUM and COL-PERI-MAX as a building block to design approximation algorithms for PERI-SUM and PERI-MAX when the general rectangular region is a square. The authors introduced an exact $O(n^2 \log n)$ algorithm for COL-PERI-SUM and two approximation algorithms for COL-PERI-MAX. The complexity status of COL-PERI-MAX remained open. Moreover, COL-ASPECT-RATIO has not been studied to this date. These methodological gaps along with the relevance of these problems for the Vietnamese land reform are a strong motivation for additional research.

3 COL-PERI-SUM can be solved in $O(n \log n)$

A polynomial-time algorithm in $O(n^2 \log n)$ for COL-PERI-SUM was proposed in Beaumont et al. (2002). In this section, we introduce a simple algorithm in $O(n \log n)$ for this problem. To that extent, we show that after ordering the rectangles’ indices by non-decreasing area, the COL-PERI-SUM problem can be reduced in $O(n)$ to the concave least-weight subsequence problem (CLWS), solvable to optimality in $O(n)$ time (Wilber 1988).

Definition 1 (Concave real-value weight function). A real-value weight function $w(i, j)$ defined for integers $0 \leq i < j \leq n$ is concave if and only if, for $0 \leq i_0 < i_1 < j_0 < j_1 \leq n$, $w(i_0, j_0) + w(i_1, j_1) \leq w(i_0, j_1) + w(i_1, j_0)$. 
Definition 2 (Concave least-weight subsequence problem). Let \( w(i, j) \) be a concave real-value weight function defined for integers \( 0 \leq i < j \leq n \). Find an integer \( k \geq 1 \) and a sequence of integers \( 0 = l_0 < l_1 < \cdots < l_{k-1} < l_k = n \) such that \( \sum_{i=0}^{k-1} w(l_i, l_{i+1}) \) is minimized.

To do this reduction, we first assume that the indices of the rectangles have been ordered in \( O(n \log n) \) by non-decreasing area: \( a_1 \leq \cdots \leq a_n \). We now highlight a property of Col-Peri-Sum which allows to focus the search on a smaller subset of solutions.

Theorem 1. Consider a solution \( s \) of Col-Peri-Sum with cost \( \Phi_1(s) \), represented as a partition \( \{S_1, \ldots, S_m\} \) of the rectangle set. Let \( i \in S_k \) and \( j \in S_l \) be two rectangles from different subsets such that \( a_i > a_j \). Any solution \( s' \) obtained by swapping these two rectangles within their respective subsets is such that:

\[
\begin{cases}
\Phi_1(s') < \Phi_1(s) & \text{if } |S_k| > |S_l| \\
\Phi_1(s') = \Phi_1(s) & \text{if } |S_k| = |S_l| \\
\Phi_1(s') > \Phi_1(s) & \text{otherwise}
\end{cases}
\]

Proof. Simply evaluate the cost difference using Equation (2):

\[
\Delta = \Phi_1(s') - \Phi_1(s) = \frac{2}{L_1} \left( |S_k| \left( a_j - a_i + \sum_{x \in S_k} a_x \right) + |S_l| \left( a_i - a_j + \sum_{x \in S_l} a_x \right) - |S_k| \sum_{x \in S_k} a_x - |S_l| \sum_{x \in S_l} a_x \right) = \frac{2}{L_1} (|S_k| - |S_l|) (a_j - a_i).
\]

This theorem defines some important features of the optimal solutions of Col-Peri-Sum:

- First, without loss of generality, any solution of Col-Peri-Sum can be presented in such a way that \( |S_1| \geq \cdots \geq |S_m| \) (re-ordering the subsets according to their cardinality).
- With this representation, if \( k < l \) and \( |S_k| = |S_l| \), there exists an optimal solution such that \( a_i \leq a_j \) for all \( i \in |S_k| \) and \( j \in |S_l| \).
- Finally, if \( k < l \) and \( |S_k| > |S_l| \), all optimal solutions satisfy \( a_i \leq a_j \) for all \( i \in |S_k| \) and \( j \in |S_l| \).

Following from these observations, there exists an optimal solution \( s^* = \{S_1, \ldots, S_m\} \) of Col-Peri-Sum such that each \( S_k \) for \( k \in \{1, \ldots, m\} \) is a subsequence (of consecutive indices) of the sequence \( \langle a_1, a_2, \ldots, a_n \rangle \). Therefore, we can find an optimal solution of Col-Peri-Sum by solving a least weight subsequence problem instance over the set of integers \( 0 \leq i < j \leq n \) with
the weight function:

$$w_P(i, j) = 2 \left( L_1 + \frac{(j - i)}{L_1} \sum_{k=i+1}^{j} a_k \right),$$

where \(w_P(i, j)\) represents the sum of the perimeters of the rectangles of indices \((i + 1, \ldots, j)\) when positioned in a single layer. Finally, Theorem 2 proves that this weight function is concave, leading to an instance of CLWS.

**Theorem 2.** The weight function \(w_P(i, j)\) is concave.

**Proof.** For \(0 \leq i_0 < i_1 < j_0 < j_1 \leq n\), one can directly verify that:

\[
\Delta' = w(i_0, j_1) + w(i_1, j_0) - w(i_0, j_0) - w(i_1, j_1)
= \frac{2}{L_1} \left( (j_1 - i_0) \sum_{x=i_0+1}^{j_1} a_x + (j_0 - i_1) \sum_{x=i_1+1}^{j_0} a_x - (j_0 - i_0) \sum_{x=i_0+1}^{j_1} a_x - (j_1 - i_1) \sum_{x=i_1+1}^{j_0} a_x \right)
= \frac{2}{L_1} \left( (i_1 - i_0) \sum_{x=j_0+1}^{i_1} a_x + (j_1 - j_0) \sum_{x=j_0+1}^{i_1} a_x \right) > 0. \square
\]

As a consequence, after prior ordering of the rectangles in \(O(n \log n)\), an optimal solution of Col-Peri-Sum can be found by solving an instance of CLWS, e.g., using the \(O(n)\) algorithm of (Wilber 1988). Col-Peri-Sum can thus be solved in \(O(n \log n)\) in the general case, and in \(O(n)\) if the rectangles are ordered by non-decreasing (or non-increasing) area in the input.

### 4 NP-hardness results

In the previous section, we have seen that an efficient \(O(n \log n)\) algorithm can be designed for Col-Peri-Sum. In contrast, we will show in the following that the “min-max” version of this problem, Col-Peri-Max, as well as the Col-Aspect-Ratio problems are more difficult.

Let Col-Peri-Max-\(\Phi\) and Col-Aspect-Ratio-\(\Phi\) be the decision problems in which one must determine whether there exists a solution of value at most \(\Phi\) for Col-Peri-Max, and Col-Aspect-Ratio, respectively. We will show that these two problems are NP-complete, by reduction from 2-Partition (Garey and Johnson 1990), hence establishing the NP-hardness of Col-Peri-Max and Col-Aspect-Ratio.
Theorem 3. Col-Peri-Max-$\Phi$ is NP-complete.

Proof. In 2-Partition, we are given $n$ positive integers $c_1, \ldots, c_n$, and should determine whether there is a partition $S_1 \cup S_2 = \{1, \ldots, n\}$, $S_1 \cap S_2 = \emptyset$ such that $\sum_{x \in S_1} c_x = \sum_{x \in S_2} c_x$.

Let $c_{\text{MAX}} = \max_{i \in \{1, \ldots, n\}} c_i$, and consider the following Col-Peri-Max-$\Phi$ instance:
- a rectangular area of length $L_1 = \frac{1}{2} \sum_{i=1}^{n} c_i$ and height $L_2 = 2c_{\text{MAX}}$;
- for $i \in \{1, \ldots, n\}$, rectangle $i$ has an area $a_i = c_i \times c_{\text{MAX}}$; and
- $\Phi = 4 \times c_{\text{MAX}}$.

Assume that 2-Partition is True: there exists a partition $(S_1, S_2)$ such that $\sum_{x \in S_1} c_x = \sum_{x \in S_2} c_x$. We build a solution of Col-Peri-Max-$\Phi$ in which the set of rectangles has been partitioned with $(S_1, S_2)$ into two layers. Each layer has the same total area, forming a solution in which all rectangles have one side of height $L_2 = 2c_{\text{MAX}}$. With this configuration, the rectangle of largest area has the largest perimeter, equal to $4 \times c_{\text{MAX}} = \Phi$, and thus Col-Peri-Max-$\Phi$ is True.

Assume that the 2-Partition instance is False. Consider a solution of Col-Peri-Max, and let $S_k$ be the layer which contains the largest rectangle with area $c_{\text{MAX}}^2$. The sum of areas in $S_k$ is different from $\frac{L_1 \times L_2}{2}$, and thus the height of this layer is different from $c_{\text{MAX}}$. Hence, the soft rectangle of area $c_{\text{MAX}}^2$ is not arranged as a square, its perimeter exceeds $4 \times c_{\text{MAX}}$, and Col-Peri-Max-$\Phi$ is False. \(\square\)

Theorem 4. Col-Aspect-Ratio-$\Phi$ is NP-complete.

Proof. As previously, consider an instance of 2-Partition with $n$ positive integers $c_1, \ldots, c_n$. Let $C = \sum_{i=1}^{n} c_i$ and $M = \frac{2(C+1)^2}{\min_{i \in \{1, \ldots, n\}} c_i}$. Define an instance of Col-Aspect-Ratio-$\Phi$ as follows:
- a rectangular area of length $L_1 = M + \frac{1}{M} + \frac{C}{2}$ and height $L_2 = 2$;
- $n$ soft rectangles with areas $c_1, \ldots, c_n$ as well as two soft rectangles of area $M$ and two soft rectangles of area $\frac{1}{M}$; and
- $\Phi = M$.

If 2-Partition is True, there exists a partition $(S_1, S_2)$ such that $\sum_{x \in S_1} c_x = \sum_{x \in S_2} c_x = \frac{C}{2}$. We build a solution of Col-Aspect-Ratio with two layers containing the rectangles of $S_1$ and $S_2$, respectively, as well as one pair of rectangles of area $M$ and $\frac{1}{M}$ each. Each layer has length $M + \frac{1}{M} + \frac{C}{2}$ and height 1. In this configuration, a maximum aspect ratio of $M$, is jointly
attained by the largest and smallest rectangles in each layer, and thus \( \text{COL-ASPECT-RATIO-} \Phi \) is \textbf{True}.

Now, assume that \( \text{2-PARTITION} \) is \textbf{False}, and discern three possible classes of solutions:

- Consider a solution of \( \text{COL-ASPECT-RATIO} \) with one layer. The rectangle of size \( \frac{1}{M} \) has an aspect ratio of \( 4M \), which exceeds \( \Phi \).

- Consider a solution of \( \text{COL-ASPECT-RATIO} \) with two or more layers, where at least one layer does not contain a rectangle of size \( M \). Let \( c \) be the area of the largest element in this layer. Two cases should be discerned:
  - If \( c = \frac{1}{M} \), then the layer contains one or two small rectangles of area \( \frac{1}{M} \), and no other rectangle. The aspect ratio of one such rectangle can be computed as the ratio of its length \( l_1 \geq \frac{1}{2}(M + \frac{1}{M} + \frac{C}{2}) \) over its height \( l_2 \leq 2 \times \frac{\frac{C}{2}}{2M + \frac{C}{2} + C} \). As such, \( \Phi \geq M \times \frac{(M + \frac{1}{M} + \frac{C}{2})^2}{4} > M \).
  - Otherwise, there exists at least one rectangle \( c_i \) in the layer, and the total area of the layer does not exceed \( C + \frac{2}{M} \). The length \( l_1 \) and height \( l_2 \) of the rectangle of area \( c_i \) satisfy \( l_1 \geq \frac{c_i}{C+\frac{1}{M}} \times (M + \frac{1}{M} + \frac{C}{2}) \) and \( l_2 \leq 2 \times \frac{C + \frac{2}{M}}{2M + \frac{C}{2} + C} \). As such, \( \Phi \geq \frac{c_i(M + \frac{1}{M} + \frac{C}{2})^2}{(C + \frac{2}{M})^2} > \frac{\min_{i=1}^{n} c_i \times M^2}{(C + 1)^2} = M \).

- Finally, consider a solution of \( \text{COL-ASPECT-RATIO} \) with two layers, where each layer contains exactly one rectangle of size \( M \). Since there is no feasible solution of \( \text{2-PARTITION} \), the total areas of the layers are different (the smaller rectangles cannot re-balance the sum due to their small area \( 1/M \)). In the layer of smallest area, the rectangle of area \( M \) has a length \( l_1 > M \) and height \( l_2 < 1 \), and thus an aspect ratio \( \Phi = \frac{x}{y} > M \).

In all cases, there is no solution with an aspect ratio smaller or equal to \( \Phi \), and thus \( \text{COL-ASPECT-RATIO-} \Phi \) is \textbf{False}. \( \square \)

5 Mixed integer programming models

Since \( \text{COL-PERI-MAX} \) and \( \text{COL-ASPECT-RATIO} \) are \textbf{NP-hard}, this section proposes mixed integer programming formulations of these problems, which can be solved to produce optimal solutions for small and medium scale instances.
These models describe a solution with \( n \) layers in which some of the layers can be empty. We associate one binary variable \( x_{ik} \) and one continuous variable \( w_{ik} \) for each rectangle \( i \) and layer \( k \). Variable \( x_{ik} \) takes value 1 if and only if rectangle \( i \) belongs to layer \( k \), and variable \( w_{ik} \) represents the length of the soft rectangle \( i \) when placed in layer \( k \), and 0 otherwise. Finally, each binary variable \( y_k \) takes value 1 if layer \( k \) is non-empty, and 0 otherwise.

### 5.1 Formulation of COL-PERI-MAX

The mathematical formulation of COL-PERI-MAX is given in Equations (5)–(17):

\[
\text{minimize } \Phi_2 \quad (5)
\]
\[
s.t. \quad 2(L_1 + L_2)(x_{ik} - 1) + 2 \left( w_{ik} + \sum_{j=1}^{n} \frac{a_j x_{jk}}{L_1} \right) \leq \Phi_2 \quad i, k \in \{1, \ldots, n\} \quad (6)
\]
\[
\sum_{k=1}^{n} x_{ik} = 1 \quad i \in \{1, \ldots, n\} \quad (7)
\]
\[
\sum_{i=1}^{n} x_{ik} \geq y_k \quad k \in \{1, \ldots, n\} \quad (8)
\]
\[
x_{ik} \leq y_k \quad i, k \in \{1, \ldots, n\} \quad (9)
\]
\[
\sum_{i=1}^{n} w_{ik} = L_1 y_k \quad k \in \{1, \ldots, n\} \quad (10)
\]
\[
w_{ik} \leq L_1 x_{ik} \quad i, k \in \{1, \ldots, n\} \quad (11)
\]
\[
a_i x_{ik} \leq L_2 w_{ik} \quad i, k \in \{1, \ldots, n\} \quad (12)
\]
\[
a_j w_{ik} - a_i w_{jk} \leq a_j L_1 (2 - x_{ik} - x_{jk}) \quad i, j, k \in \{1, \ldots, n\}, i \neq j \quad (13)
\]
\[
a_i w_{jk} - a_j w_{ik} \leq a_i L_1 (2 - x_{ik} - x_{jk}) \quad i, j, k \in \{1, \ldots, n\}, i \neq j \quad (14)
\]
\[
x_{ji} \in \{0, 1\} \quad i, j \in \{1, \ldots, n\} \quad (15)
\]
\[
w_{ik} \geq 0 \quad i, k \in \{1, \ldots, n\} \quad (16)
\]
\[
y_k \in \{0, 1\} \quad k \in \{1, \ldots, n\} \quad (17)
\]
\[
\Phi_2 \geq 0 \quad (18)
\]

Constraints (7)–(9) ensure that every rectangle is included in a layer and that \( y_k \) takes value 1 when at least one rectangle is included in layer \( k \). Constraints (10) state that the sum of the length of the rectangles of each layer \( k \) equals \( L_1 \) if this layer is used \( (y_k = 1) \), and 0 otherwise.
otherwise. Constraints (11) and (12) impose that \((w_{ik} = 0) \iff (x_{ik} = 0)\). Finally, Constraints (13) and (14) ensure that if two rectangles \(i\) and \(j\) are in the same layer \(k\), then \(a_i/w_{ik} = a_j/w_{jk}\).

This formulation can be strengthened with the addition of some simple optimality cuts which eliminate symmetrical solutions:

\[
\begin{align*}
y_k &\geq y_{k+1} & k &\in \{1, \ldots, n-1\} \\
x_{ik} &= 0 & i &\in \{1, \ldots, n\}, k &\in \{i+1, \ldots, n\}
\end{align*}
\] (19) (20)

The first set of constraints forces the use of layers according to the order of their indices, while the second set of constraints forces any rectangle \(i\) to belong to a layer of index \(k \in \{1, \ldots, i\}\).

### 5.2 Formulation of COL-ASPECT-RATIO

The objective function \(\Phi_3\) is nonlinear, and we did not find a direct mixed integer programming formulation of COL-ASPECT-RATIO. Instead, we propose two alternative approaches to generate optimal solutions for this problem. The first approach relies on a change of objective which leads to a linear formulation returning the same optimal solutions as COL-ASPECT-RATIO. The second approach exploits the fact that the decision problem COL-ASPECT-RATIO-\(\Phi\) can be formulated as a MIP. Solving this subproblem in a binary search allows to solve the original optimization problem.

#### 5.2.1 First approach – Change of objective function

We introduce an alternative objective function \(\Phi_4\) for COL-ASPECT-RATIO which can be modeled in a linear formulation. This objective function can be computed as:

\[
\Phi_4 = \max_{k} \max_{i \in S_k} \frac{|w(S_k) - \frac{a_i}{w_i(S_k)}|}{\sqrt{a_i}}.
\] (21)

The following lemma will be used to prove the equivalence between the two objectives:

**Lemma 1.** Given two soft rectangles \(i\) and \(j\) with side lengths \((l_i, h_i)\) and \((l_j, h_j)\), we have

\[
\frac{\max(l_i, h_i)}{\min(l_i, h_i)} \geq \frac{\max(l_j, h_j)}{\min(l_j, h_j)} \iff \frac{|l_i - h_i|}{\sqrt{l_i h_i}} \geq \frac{|l_j - h_j|}{\sqrt{l_j h_j}}.
\]
Proof. Without loss of generality, we can assume that \( l_i \geq h_i \) and \( l_j \geq h_j \). Then,

\[
\frac{\max(l_i, h_i)}{\min(l_i, h_i)} \geq \frac{\max(l_j, h_j)}{\min(l_j, h_j)}
\]

\[
\iff \frac{l_i}{h_i} \geq \frac{l_j}{h_j}
\]

\[
\iff \sqrt{\frac{l_i}{h_i}} \geq \sqrt{\frac{l_j}{h_j}}
\]

\[
\iff \left( \sqrt{\frac{l_i}{h_i}} - \sqrt{\frac{l_j}{h_j}} \right) \left( 1 + \frac{1}{\sqrt{\frac{l_i}{h_i}} \sqrt{\frac{l_j}{h_j}}} \right) \geq 0
\]

\[
\iff \sqrt{\frac{l_i}{h_i}} - \sqrt{\frac{l_j}{h_j}} \geq \sqrt{\frac{l_j}{h_j}} - \sqrt{\frac{l_i}{h_i}}
\]

\[
\iff \frac{l_i - h_i}{\sqrt{l_i h_i}} \geq \frac{l_j - h_j}{\sqrt{l_j h_j}}
\]

\[
\iff \frac{|l_i - h_i|}{\sqrt{l_i h_i}} \geq \frac{|l_j - h_j|}{\sqrt{l_j h_j}}
\]

\[
\iff |l_i - h_i| \sqrt{l_i h_i} \geq |l_j - h_j| \sqrt{l_j h_j}
\]

Theorem 5. Let \( s_3 \) and \( s_4 \) be two optimal solutions obtained with objectives \( \Phi_3 \) and \( \Phi_4 \), respectively. Then, \( \Phi_3(s_3) = \Phi_3(s_4) \), \( \Phi_4(s_3) = \Phi_4(s_4) \), and \( s_3 \) and \( s_4 \) are optimal for the objectives \( \Phi_4 \) and \( \Phi_3 \), respectively.

Proof. For any solution \( s \), as a consequence of Lemma 1, if \( \Phi_4(s) \) attains its minimum for a rectangle \( i \in \{1, \ldots, n\} \) of length \( l_i \) and height \( h_i \), then \( \Phi_3(s) \) attains its minimum for the same rectangle, and vice-versa. Therefore \( \Phi_4(s) = \frac{|l_x - h_x|}{\sqrt{l_x h_x}} \) and \( \Phi_3(s) = \frac{\max(l_x, h_x)}{\min(l_x, h_x)} \).

Now, assume that \( \Phi_4(s_4) \) and \( \Phi_3(s_3) \) attain their minimum for rectangles \( x \) and \( y \), respectively. Therefore,

\[
\Phi_4(s_4) = \frac{|l_x - h_x|}{\sqrt{l_x h_x}}, \quad \Phi_3(s_4) = \frac{\max(l_x, h_x)}{\min(l_x, h_x)}
\]

\[
\Phi_4(s_3) = \frac{|l_y - h_y|}{\sqrt{l_y h_y}}, \quad \Phi_3(s_3) = \frac{\max(l_y, h_y)}{\min(l_y, h_y)}
\]

Since \( s_4 \) is an optimal solution for objective \( \Phi_4 \), \( \Phi_4(s_4) \leq \Phi_4(s_3) \) and:

\[
\frac{|l_x - h_x|}{\sqrt{l_x h_x}} \leq \frac{|l_y - h_y|}{\sqrt{l_y h_y}}
\]
Therefore, as a consequence of Lemma 1, we have
\[
\frac{\max(l_x, h_x)}{\min(l_x, h_x)} \leq \frac{\max(l_y, h_y)}{\min(l_y, h_y)}
\]
Similarly, since \(s_3\) is an optimal solution for objective \(\Phi_3\), \(\Phi_3(s_3) \leq \Phi_3(s_4)\) and:
\[
\frac{\max(l_y, h_y)}{\min(l_y, h_y)} \leq \frac{\max(l_x, h_x)}{\min(l_x, h_x)}
\]
Overall,
\[
\Phi_3(s_3) = \frac{\max(l_y, h_y)}{\min(l_y, h_y)} = \frac{\max(l_x, h_x)}{\min(l_x, h_x)} = \Phi_3(s_4),
\]
and \(s_4\) is an optimal solution for objective \(\Phi_3\). A similar proof shows that \(s_3\) is an optimal solution for objective \(\Phi_4\).

Based on this change of objective function, \textsc{Col-Aspect-Ratio} can be formulated as:
\[
\text{min } \Phi \\
\text{s.t. } \text{Constraints (7)} - (15)
\]
\[
\delta_{ik} + L_1(1 - x_{ik}) \geq w_{ik} - \sum_{j=1}^{n} \frac{a_j x_{jk}}{L_1} \quad i, k \in \{1, \ldots, n\} \quad (22)
\]
\[
\delta_{ik} + L_2(1 - x_{ik}) \geq -w_{ik} + \sum_{j=1}^{n} \frac{a_j x_{jk}}{L_1} \quad i, k \in \{1, \ldots, n\} \quad (23)
\]
\[
\Phi \geq \frac{\delta_{ik}}{\sqrt{a_i}} \quad i, k \in \{1, \ldots, n\} \quad (24)
\]
\[
\delta_{ik} \geq 0 \quad i, k \in \{1, \ldots, n\}. \quad (25)
\]
Solving this formulation to optimality generates an optimal solution of \textsc{Col-Aspect-Ratio}.

The value of this solution must be recomputed a-posteriiori according to the original objective.

The model uses \(n^2\) additional continuous variables \(\delta_{ik}\), as well as a continuous variable \(\Phi\) representing the value of the alternative objective function. According to Constraints (22) and (23), if a rectangle \(i\) is in layer \(S_k\), then \(\delta_{ik} = |l_i - h_i|\) where \(l_i\) and \(h_i\) represent the length and height of the rectangle in the current solution, otherwise \(\delta_{ik} = 0\).
5.2.2 Second approach – Binary search

Another solution approach consists in modeling the decision problem \textsc{Col-Aspect-Ratio-\Phi} as a MIP. In this case, a maximum aspect ratio \Phi is set as a constraint, and the goal is to find a feasible solution. The feasibility model can be written as follows:

Constraints (7)–(15)

\begin{align}
L_1h_k &= \sum_{i=1}^{n} a_i x_{ik} \quad k \in \{1,\ldots,n\} \quad (26) \\
w_{ik} &\leq \Phi h_k \quad i, k \in \{1,\ldots,n\} \quad (27) \\
h_k &\leq \Phi w_{ik} \quad i, k \in \{1,\ldots,n\} \quad (28) \\
h_k &\in \mathbb{R} \quad k \in \{1,\ldots,n\} \quad (29)
\end{align}

In this model, each variable \( h_k \) for \( k \in \{1,\ldots,n\} \) stores the height of layer \( S_k \), and Constraints (27)–(28) force the aspect ratio to be no higher than \( \Phi \). To solve the original optimization problem, we do a binary search over \( \Phi \) and solve \textsc{Col-Aspect-Ratio-\Phi} at each step. The starting interval is set to \([\Phi^{\text{low}}, \Phi^{\text{up}}]\) where \( \Phi^{\text{low}} = 1 \) and \( \Phi^{\text{up}} = \Phi_3(s_1) \), where \( s_1 \) is an optimal solution of \textsc{Col-Peri-Sum} found in \( O(n \log n) \) time. The binary search stops as soon as \( \Phi^{\text{up}} - \Phi^{\text{low}} < 0.01 \).

6 Computational Experiments

To complete the theoretical results of this article, we conducted computational experiments to evaluate the efficiency of the solution methods for the three problems and compare their solutions. All algorithms were implemented in C++ and the mathematical models were solved with CPLEX version 12.4. The experiments were performed on a single thread of an Intel i7-3615QM 2.3 GHz CPU with 10GB RAM, running Mac OS Sierra version 10.12.6, and subject to a CPU time limit of one hour for each run.

We randomly generated benchmark instances with \( n \in \{10, 15, 20, 25, 30, 35, 40\} \) soft rectangles. These instances are subdivided into three classes, and three instances were generated for each class and size for a total of 63 instances.

- **Class U** – The area of each item is sampled in a uniform distribution: \( X \sim U(1, 200) \).
• **Class MU** – The area of each item is sampled in a mixture of three uniform distributions:
  \[ X \sim \frac{1}{3} [\mathcal{U}(1, 10) + \mathcal{U}(11, 50) + \mathcal{U}(51, 150)] \].

• **Class MN** – The area of each item is sampled in a mixture of three normal distributions:
  \[ X \sim \frac{1}{3} [\mathcal{N}(5, 2) + \mathcal{N}(25, 10) + \mathcal{N}(125, 50)] \], but another sample is taken whenever the area is larger than 200.

Finally, the dimensions of the hard rectangle are generated as follows for each instance. Let \( A \) be the sum of the areas of the soft rectangles. Length \( L_1 \) is randomly generated with uniform probability in \( \{\lceil \sqrt{A/3} \rceil, \ldots, \lfloor \sqrt{3A} \rfloor \} \). The length of the other side is set to \( L_2 = \lfloor A/L_1 \rfloor \). Then, \( A - L_1L_2 \) soft rectangles are randomly selected, and the area of each rectangle is reduced by one unit in such a way that, after reduction, the area of the hard rectangle coincides with the sum of the areas of the soft rectangles. All benchmark instances are available at [https://w1.cirrelt.ca/~vidalt/en/research-data.html](https://w1.cirrelt.ca/~vidalt/en/research-data.html).

### 6.1 Performance analysis

This section compares the CPU time needed to solve \textsc{Col-Peri-Sum}, \textsc{Col-Peri-Max}, and \textsc{Col-Aspect-Ratio}. As expected, the solution of \textsc{Col-Peri-Sum} in \( \mathcal{O}(n \log n) \) is extremely fast, with a measured CPU time of the order of a few milliseconds for all considered instances, such that we concentrate our analyzes on the mathematical programming algorithm for \textsc{Col-Peri-Max} as well as the reformulation and binary search approaches for \textsc{Col-Aspect-Ratio}. To speed up the solution methods, we always generate the optimal solution of \textsc{Col-Peri-Sum} and use it as an initial feasible solution.

Tables 1 to 3 report, for each instance class and algorithm, the number of nodes in the search tree (\textsc{Nodes}), the CPU time in seconds (\textsc{Time}), as well as the best found lower bound (\textsc{LB}) and upper bound (\textsc{UB}). For the reformulation-based approach for \textsc{Col-Aspect-Ratio}, columns \textsc{LB}_4 and \textsc{UB}_4 correspond to objective \( \Phi_4 \), and the value of the primal solution for objective \( \Phi_3 \) is indicated in column \textsc{UB}_3. TL in column \textsc{Time} means that the CPU time limit of 3600 seconds has been attained. Finally, for the binary search approach for \textsc{Col-Aspect-Ratio}, we indicate the number of completed iterations in column \textsc{It}_{\text{BS}}.

As observed in these experiments, the proposed MIP models can be solved to optimality for all benchmark instances with 10 soft rectangles, as well as a few instances with up to 30 rectangles for \textsc{Col-Peri-Max} and 40 rectangles for \textsc{Col-Aspect-Ratio}. Yet, the number of
### Table 1: Class U – Performance comparisons

| Data | Size | Col-Peri-Max | Col-Aspect-Ratio – Reformulation | Col-Aspect-Ratio – B. Search |
|------|------|--------------|----------------------------------|-------------------------------|
|      |      | Nodes Time LB UB | Nodes Time LB UB | Itbs Time LB UB |
| p01  | 10   | 547 0.53 52.53 | 176 0.39 0.95 0.95 | 8 1.21 2.51 2.51 |
| p02  | 10   | 27 0.20 55.17 | 68 0.22 1.72 1.72 | 10 1.38 4.75 4.75 |
| p03  | 10   | 101 0.20 55.54 | 1 0.08 1.05 1.05 | 8 0.93 2.72 2.73 |
| p04  | 15   | 5413k TL 52.27 | 69 0.74 0.82 0.82 | 8 4.91 2.22 2.22 |
| p05  | 15   | 38k 92.46 | 193 1.85 0.99 0.99 | 9 4.27 2.59 2.60 |
| p06  | 15   | 8054k TL 48.63 | 640k 479.41 1.98 1.98 | 10 560.23 5.75 5.76 |
| p07  | 20   | 351k TL 43.27 | 308 3.38 2.60 2.60 | 12 29.04 8.63 8.64 |
| p08  | 20   | 5139k TL 52.27 | 293k TL 0.00 0.86 2.30 | 2 TL 2.20 3.41 |
| p09  | 20   | 38k 92.46 | 720 1.67 2.20 2.20 | 11 12.05 6.66 6.67 |
| p10  | 25   | 1 0.49 56.17 | 714k TL 0.00 1.79 5.00 | 2 TL 1.00 5.34 |
| p11  | 25   | 472k TL 36.01 | 150k TL 0.00 1.47 3.92 | 0 TL 1.00 8.34 |
| p12  | 25   | 109k TL 41.35 | 149k TL 0.34 2.59 8.58 | 1 TL 1.00 28.70 |
| p13  | 30   | 74k TL 19.30 | 108k TL 0.64 1.42 3.76 | 1 TL 1.00 5.86 |
| p14  | 30   | 325k TL 21.43 | 1743k TL 0.00 1.40 3.68 | 1 TL 1.00 7.02 |
| p15  | 30   | 196k TL 42.46 | 179k TL 1.28 1.96 5.07 | 2 TL 1.00 6.66 |
| p16  | 35   | 66k 285.98 | 142k 550.19 2.18 2.18 | 12 2355.92 6.61 6.61 |
| p17  | 35   | 688k TL 40.79 | 281k TL 0.76 1.33 3.48 | 3 TL 2.98 3.96 |
| p18  | 35   | 204k TL 21.41 | 220k TL 1.47 2.24 6.86 | 2 TL 1.00 12.49 |
| p19  | 40   | 267k TL 32.40 | 225k TL 1.77 2.13 6.39 | 5 TL 5.72 6.66 |
| p20  | 40   | 68k 44.12 | 150k TL 0.91 1.86 5.27 | 2 TL 1.00 7.10 |
| p21  | 40   | 53k TL 44.12 | 68k TL 0.05 2.73 9.35 | 2 TL 1.00 21.33 |

### Table 2: Class MU – Performance comparisons

| Data | Size | Col-Peri-Max | Col-Aspect-Ratio – Reformulation | Col-Aspect-Ratio – B. Search |
|------|------|--------------|----------------------------------|-------------------------------|
|      |      | Nodes Time LB UB | Nodes Time LB UB | Itbs Time LB UB |
| p01  | 10   | 179 0.31 55.29 | 17 0.24 1.39 1.39 | 9 1.03 3.64 3.65 |
| p02  | 10   | 44 0.07 55.78 | 293k TL 0.00 0.86 2.30 | 2 TL 2.20 3.41 |
| p03  | 10   | 4k 1.22 52.76 | 64 0.18 1.97 1.97 | 9 0.72 5.70 5.71 |
| p04  | 15   | 318k TL 40.41 | 179k TL 0.22 1.79 5.01 | 3 TL 1.00 7.22 |
| p05  | 15   | 182k TL 33.64 | 328k TL 1.71 2.59 8.57 | 3 TL 1.00 17.06 |
| p06  | 15   | 389k TL 40.00 | 714k TL 0.00 1.79 5.00 | 2 TL 1.00 5.34 |
| p07  | 20   | 247k TL 36.01 | 150k TL 0.00 1.47 3.92 | 0 TL 1.00 8.34 |
| p08  | 20   | 109k TL 41.35 | 149k TL 0.34 2.59 8.58 | 1 TL 1.00 28.70 |
| p09  | 20   | 47k TL 19.30 | 108k TL 0.64 1.42 3.76 | 1 TL 1.00 5.86 |
| p10  | 20   | 101k TL 44.48 | 54k TL 0.12 1.09 2.83 | 1 TL 1.93 2.86 |
| p11  | 20   | 53k TL 44.12 | 68k TL 0.05 2.73 9.35 | 2 TL 1.00 21.33 |
search nodes and CPU time grow very quickly with the number of soft rectangles \( n \). Despite the symmetry-breaking inequalities, some instances with 15 rectangles lead to over a million search nodes. The reformulation approach and the binary search approach for \textup{Col-Aspect-Ratio} find 31/63 and 30/63 optimal solutions, respectively. The reformulation approach is generally faster than the binary search algorithm for small instances. Yet, a drawback of this algorithm is that it searches for an optimal solution according to objective \( \Phi_4 \). When optimality is attained, this solution is optimal for \( \Phi_3 \) due to Theorem 5. When an optimality gap remains, the primal solution obtained from the algorithm gives a valid upper bound for objective \( \Phi_3 \), but the dual information (and performance guarantee) is lost. Finally, we did not observe a significant difference of performance when comparing the results of the three instance classes (\( \textup{U}, \textup{MU} \) and \( \textup{MN} \)). We noted that two larger instances with 35 and 40 rectangles were solved to optimality for class \( \textup{MU} \), a phenomenon which did not happen for \( \textup{U} \) and \( \textup{MN} \).

In general, the difficulties encountered when solving instances with 10 to 40 rectangles already show the limitations of available mathematical programming algorithms for \textup{Col-Peri-Max} and \textup{Col-Aspect-Ratio}. Future progress on exact approaches for NP-hard problems

| Data | Size | COL-PERI-MAX | COL-ASPECT-RATIO – Reformulation | COL-ASPECT-RATIO – B. Search |
|------|------|--------------|----------------------------------|-----------------------------|
|      |      | Nodes Time LB UB | Nodes Time LB4 UB4 UB3 | ItBS Time LB UB |
| p01  | 10   | 386 0.31 50.83 50.83 | 10 0.27 1.05 1.05 2.75 | 9 0.87 2.74 2.75 |
| p02  | 10   | 62 0.10 51.50 51.50 | 1 0.06 0.82 0.82 2.22 | 7 0.41 2.21 2.22 |
| p03  | 10   | 267 0.27 51.00 51.00 | 33 0.28 1.94 1.94 5.60 | 9 0.75 5.59 5.60 |
| p04  | 15   | 38k 20.37 39.80 39.80 | 13k 18.08 1.55 1.55 4.17 | 9 147.71 4.17 4.17 |
| p05  | 15   | 130k 94.84 40.99 40.99 | 2k 5.06 1.03 1.03 2.68 | 8 20.51 2.67 2.68 |
| p06  | 15   | 4971k TL 44.67 45.96 | 27 0.86 1.65 1.65 4.49 | 9 5.18 4.49 4.50 |
| p07  | 20   | 364 9.13 53.17 53.17 | 23k 200.99 2.82 2.82 9.83 | 11 2866.39 9.83 9.83 |
| p08  | 20   | 6k 35.00 53.38 53.38 | 31k 484.53 2.06 2.06 6.09 | 10 259.48 6.08 6.09 |
| p09  | 20   | 1427k TL 36.61 54.55 | 59k 290.84 1.05 1.05 2.73 | 8 554.92 2.73 2.73 |
| p10  | 25   | 2188k TL 29.04 43.08 | 987k TL 0.56 0.67 1.94 | 4 TL 1.76 1.95 |
| p11  | 25   | 233k TL 40.91 54.55 | 156 11.20 2.49 2.49 8.09 | 10 76.60 8.08 8.09 |
| p12  | 25   | 1341k TL 53.67 53.67 | 429k TL 0.55 0.93 2.45 | 8 661.27 2.44 2.45 |
| p13  | 30   | 7k 139.13 54.12 54.12 | 285k TL 1.39 2.08 6.18 | 5 TL 5.47 6.21 |
| p14  | 30   | 843k TL 35.00 54.41 | 316k TL 0.80 1.18 3.06 | 2 TL 2.28 3.55 |
| p15  | 30   | 3k 42.77 51.23 51.23 | 105k TL 1.85 2.81 9.81 | 2 TL 6.85 12.69 |
| p16  | 35   | 113k TL 39.18 55.43 | 87k TL 0.14 1.82 5.12 | 1 TL 1.00 14.14 |
| p17  | 35   | 1526k TL 40.41 53.33 | 272k TL 0.14 0.56 1.73 | 2 TL 1.55 1.74 |
| p18  | 35   | 160k TL 44.24 52.57 | 80k TL 0.55 1.91 5.48 | 1 TL 1.00 9.88 |
| p19  | 40   | 137k TL 34.26 56.43 | 49k TL 0.59 1.93 5.53 | 0 TL 1.00 6.92 |
| p20  | 40   | 294k TL 30.57 55.28 | 251k TL 0.84 1.00 2.62 | 3 TL 2.31 2.64 |
| p21  | 40   | 141k TL 9.94 54.55 | 137k TL 1.09 1.71 4.70 | 0 TL 1.00 5.09 |
may allow to solve larger instances in the future, but years of research may be needed before handling more realistic instances with over a hundred rectangles. Alternatively, heuristics and metaheuristics could be used to solve larger problems. As we noted a complexity gap between Col-Peri-Sum and the other two problems, in spite of the relations between the three objectives, we are interested to see if the solution of Col-Peri-Sum can constitute a viable heuristic for the two more difficult objectives. This is the focus of the next section.

6.2 Solution evaluations in relation to other objectives

The objective functions of the three considered problems are different but not strongly conflicting. Yet, Col-Peri-Sum can be solved in $O(n \log n)$, while Col-Peri-Max and Col-Aspect-Ratio are NP-hard. In this last analysis, we investigate how close these problems are from each other in practice. This is achieved by evaluating the optimal solution of one problem according to the objective function of each other. In particular, we are interested to see if the solution of Col-Peri-Sum can be used as a simple heuristic for Col-Peri-Max and Col-Aspect-Ratio.

For this analysis, we gathered all instances that are solved to optimality for all three problems: all instances with $n = 10$; instances U-p05, MU-p04, MU-p06, MN-p04, and MN-p05 with $n = 15$; and instances U-p09, MU-p09, MN-p07, and MN-p08 with $n = 20$. For each objective $\Phi_x$, we evaluated the quality $\Phi_y(s_x^*)$ of its optimal solution $s_x^*$ relatively to each other objective $y \in \{1, 2, 3\}$, and report the results as the performance ratio $\Phi_y(s_x^*)/\Phi_y(s_y^*)$ in Table 4.

These experiments first confirm the fact that the three objectives produce significantly different solutions. For these instances, the optimal solutions of Col-Peri-Sum are within an average factor of 1.05 of the optimal solutions of Col-Peri-Max when evaluated according to objective $\Phi_2$, and are better than the optimal solutions of Col-Aspect-Ratio with a factor of 1.11. Similarly, the optimal solutions of Col-Peri-Sum give a better approximation of the optimal solutions of Col-Aspect-Ratio than the optimal solutions of Col-Peri-Max (with a factor of 1.50 compared to 14.01).

One likely explanation of these observations is that the objective of Col-Peri-Max mainly concentrates the optimization on rectangles of large area, so as to minimize their perimeter. In Col-Peri-Max, small rectangles almost never play a role as they are unlikely to realize the maximum. In Col-Aspect-Ratio, in contrast, small and large rectangles are equally
Table 4: Optimal solutions for one objective evaluated according to the other objectives

| Evaluated as | Col-Peri-Sum – Φ₁ | Col-Peri-Max – Φ₂ | Col-Aspect-Ratio – Φ₃ |
|--------------|--------------------|--------------------|-----------------------|
| Solved as    | Φ₁  | Φ₂  | Φ₃  | Φ₁  | Φ₂  | Φ₃  | Φ₁  | Φ₂  | Φ₃  |
| MN-p01       | 1.00 | 1.30 | 1.00 | 1.12 | 1.00 | 1.07 | 1.60 | 13.09 | 1.00 |
| MN-p02       | 1.00 | 1.11 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 22.02 | 1.00 |
| MN-p03       | 1.00 | 1.55 | 1.00 | 1.04 | 1.00 | 1.04 | 1.00 | 12.86 | 1.00 |
| U-p01        | 1.00 | 1.06 | 1.00 | 1.08 | 1.00 | 1.09 | 1.01 | 4.27  | 1.00 |
| U-p02        | 1.00 | 1.00 | 1.13 | 1.00 | 1.00 | 1.31 | 1.60 | 1.60  | 1.00 |
| U-p03        | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00  | 1.00 |
| MU-p01       | 1.00 | 1.23 | 1.05 | 1.05 | 1.00 | 1.13 | 1.07 | 5.97  | 1.00 |
| MU-p02       | 1.00 | 1.06 | 1.01 | 1.00 | 1.00 | 1.00 | 1.00 | 22.69 | 1.00 |
| MU-p03       | 1.00 | 1.19 | 1.00 | 1.05 | 1.00 | 1.05 | 1.00 | 7.09  | 1.00 |
| MN-p04       | 1.00 | 1.41 | 1.08 | 1.00 | 1.00 | 1.09 | 1.08 | 38.85 | 1.00 |
| MN-p05       | 1.00 | 1.24 | 1.02 | 1.00 | 1.00 | 1.04 | 1.00 | 9.76  | 1.00 |
| U-p05        | 1.00 | 1.04 | 1.02 | 1.01 | 1.00 | 1.03 | 1.67 | 2.31  | 1.00 |
| MU-p04       | 1.00 | 1.11 | 1.02 | 1.13 | 1.00 | 1.06 | 2.24 | 21.59 | 1.00 |
| MU-p06       | 1.00 | 1.27 | 1.01 | 1.00 | 1.00 | 1.00 | 1.24 | 29.31 | 1.00 |
| MN-p07       | 1.00 | 1.15 | 1.25 | 1.24 | 1.00 | 1.55 | 1.16 | 14.65 | 1.00 |
| MN-p08       | 1.00 | 1.23 | 1.14 | 1.15 | 1.00 | 1.33 | 1.22 | 9.31  | 1.00 |
| U-p09        | 1.00 | 1.06 | 1.00 | 1.06 | 1.00 | 1.10 | 1.37 | 6.35  | 1.00 |
| MU-p09       | 1.00 | 1.24 | 1.04 | 1.00 | 1.00 | 1.07 | 5.74 | 29.47 | 1.00 |
| Average      | 1.00 | 1.18 | 1.04 | 1.05 | 1.00 | 1.11 | 1.50 | 14.01 | 1.00 |

important, since the maximum aspect ratio can be attained regardless of the rectangle area. Finally, Col-Peri-Sum must optimize the perimeter of all rectangles, regardless of their area, so as to minimize the total sum. This objective leads to optimal solutions which tend to have good overall aspect ratios, regardless of rectangle size.

Finally, a precise analysis of the solutions shows that Col-Peri-Sum produced five optimal solutions for Col-Peri-Max and six optimal solutions for Col-Aspect-Ratio over 18 instances. In one exceptional case (instance p03 of class U), the three methods converged towards the same optimal solution. This situation happened because the optimal solution contained a single layer, but other situations can lead to this behavior: e.g., if a feasible solution exists in which all soft rectangles take the shape of a square, then this solution is indeed optimal for the three objectives.

7 Conclusions

In this paper, we investigated three soft rectangle packing problems: Col-Peri-Sum, Col-Peri-Max and Col-Aspect-Ratio. The effective resolution of these problems is of foremost
importance for the ongoing land-allocation reform in Vietnam. The objectives considered in these problems model different aspects of fairness and wasted-land minimization. We introduced an $O(n \log n)$ exact algorithm for \textsc{Col-Peri-Sum}. Then, we demonstrated that the two others problems are NP-hard, and proposed compact MIP formulations to solve them. In the case of \textsc{Col-Aspect-Ratio}, an objective reformulation and a binary search scheme were proposed to overcome non-linearities. Through a set of experimental analyzes on 63 benchmark instances, we observed that the resolution of the MIP formulations is currently practicable for problem instances involving 10 to 40 soft rectangles. To solve larger instances of \textsc{Col-Peri-Max} and \textsc{Col-Aspect-Ratio}, we may use the $O(n \log n)$-time algorithm for \textsc{Col-Peri-Sum} as a simple heuristic.

The research perspectives are numerous. The proposed MIP formulations can possibly be improved with additional valid inequalities or optimality cuts, and the set-partitioning formulation of the problem can certainly be exploited to develop efficient branch-and-price algorithms. Metaheuristics could also be developed to provide solutions for larger instances or integrate additional restrictions or objectives. Finally, whether \textsc{Col-Peri-Max} and \textsc{Col-Aspect-Ratio} are strongly NP-hard remains an interesting open question.

References

Arimoto, Y. 2010. Impact of land readjustment project on farmland use and structural adjustment: The case of Niigata, Japan. \textit{Agricultural and Applied Economics Association 2010 AAEA, CAES, WAEA Joint Annual Meeting.}. Denver, USA.

Beaumont, O., V. Boudet, F. Rastello, Y. Robert. 2001. Matrix multiplication on heterogeneous platforms. \textit{IEEE Transactions on Parallel and Distributed Systems} \textbf{12}(10) 1033–1051.

Beaumont, O., V. Boudet, F. Rastello, Y. Robert. 2002. Partitioning a square into rectangles: NP-Completeness and approximation algorithms. \textit{Algorithmica} \textbf{34}(3) 217–239.

Borgwardt, S., A. Brieden, P. Gritzmann. 2011. Constrained minimum-k-star clustering and its application to the consolidation of farmland. \textit{Operational Research} \textbf{11}(1) 1–17.

Borgwardt, S., A. Brieden, P. Gritzmann. 2014. Geometric clustering for the consolidation of farmland and woodland. \textit{The Mathematical Intelligencer} \textbf{36}(2) 37–44.

Brieden, A., P. Gritzmann. 2004. A quadratic optimization model for the consolidation of farmland by means of lend-lease agreements. D. Ahr, R. Fahrion, M. Oswald, G. Reinelt, eds., \textit{Operations Research Proceedings 2003}. Springer Berlin Heidelberg, 324–331.
Bui, Q. T., Q. D. Pham, Y. Deville. 2013. Solving the agricultural land allocation problem by constraint-based local search. C. Schulte, ed., Principles and Practice of Constraint Programming, Lecture Notes in Computer Science, vol. 8124. Springer Berlin Heidelberg, 749–757.

Cay, T., T. Ayten, F. Iscan. 2006. An investigation of reallocation model based on interview in land consolidation. Proceeding of the 23rd FIG Congress, Shaping the Change. Munich, Germany.

Cay, T., T. Ayten, F. Iscan. 2010. Effects of different land reallocation models on the success of land consolidation projects: Social and economic approaches. Land Use Policy 27(2) 262–269.

Cay, T., M. Uyan. 2013. Evaluation of reallocation criteria in land consolidation studies using the analytic hierarchy process (AHP). Land Use Policy 30(1) 541–548.

Demetriou, D., L. See, J. Stillwell. 2013. A spatial genetic algorithm for automating land partitioning. International Journal of Geographical Information Science 27 2391–2409.

Fügenschuh, A, K. Junosza-Szaniawski, Z. Lonc. 2014. Exact and approximation algorithms for a soft rectangle packing problem. Optimization 63(11) 1637–1663.

Garey, M. R., D. S. Johnson. 1990. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA.

General Statistical Office. 2004. Statistical yearbook 2004. Statistical Publishing House.

Gliesch, A., M. Ritt, M. C. O. Moreira. 2017. A genetic algorithm for fair land allocation. Proceedings of the Genetic and Evolutionary Computation Conference. GECCO ’17, ACM, New York, NY, USA, 793–800.

Hakli, H., H. Uuz. 2017. A novel approach for automated land partitioning using genetic algorithm. Expert Systems with Applications 82(1) 10–18.

Heltberg, R. 1998. Rural market imperfections and the farm size-productivity relationship: Evidence from Pakistan. World Development 26 1807–1826.

Huang, Q., M. Li, Z. Chen, F. Li. 2011. Land consolidation: An approach for sustainable development in rural China. AMBIO 40(1) 93–95.

Ibaraki, T., K. Nakamura. 2006. Packing problems with soft rectangles. F. Almeida, M. J. Blesa Aguilera, C. Blum, J. M. Moreno Vega, M. Pérez Pérez, A. Roli, M. Sampels, eds., Hybrid Metaheuristics. Springer Berlin Heidelberg, Berlin, Heidelberg, 13–27.

Ji, P., K. He, Y. Jin, H. Lan, C. Li. 2017. An iterative merging algorithm for soft rectangle packing and its extension for application of fixed-outline floorplanning of soft modules. Computers & Operations Research 86 110–123.

Journal of Rural Economy. 2008. Agricultural mechanization: When does it go over the start? http://ipsard.gov.vn/mobile/tID2264_Co-gioi-hoa-nong-nghep-Khi-nao-qua-buoc-khoi-dong-.html. [Online; accessed 01-May-2018].
Kong, T. Y., D. M. Mount, M. Werman. 1987. The decomposition of a square into rectangles of minimal perimeter. *Discrete Applied Mathematics* **16**(3) 239–243.

Kong, T. Y., D. M. Mount, M. Werman. 1988. The decomposition of a rectangle into rectangles of minimal perimeter. *SIAM Journal on Computing* **17**(6) 1215–1231.

Lam, M-L. 2001. Land fragmentation—a constraint for Vietnam agriculture. *Vietnam’s socio-economic development* **26** 73–80.

March, S. P., T. G. MacAulay. 2006. Farm size and land use change in Vietnam following land reforms. *47th Annual Conference of the Australian Agricultural and Resource Economics Society*. Fremantle, Australia.

Nagamochi, H., Y. Abe. 2007. An approximation algorithm for dissecting a rectangle into rectangles with specified areas. *Discrete Applied Mathematics* **155**(4) 523–537.

Paes, F.G., A.A. Pessoa, T. Vidal. 2017. A hybrid genetic algorithm with decomposition phases for the unequal area facility layout problem. *European Journal of Operational Research* **256**(3) 742–756.

Pham, V-H., G. T. MacAulay, S. P. Marsh. 2007. The economics of land fragmentation in the north of Vietnam*. *Australian Journal of Agricultural and Resource Economics* **51**(2) 195–211.

Sundqvist, P., L. Anderson. 2006. A study of the impacts of land fragmentation on agricultural productivity in Northern Vietnam. Bachelor’s thesis, Uppsala University, Sweden.

Wilber, R. 1988. The concave least-weight subsequence problem revisited. *Journal of Algorithms* **9**(3) 418–425.

Young, F. Y., C. C. N. Chu, W. S. Luk, Y. C. Wong. 2001. Handling soft modules in general nonslicing floorplan using lagrangian relaxation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **20**(5) 687–692.