Journal of Global Optimization
https://doi.org/10.1007/s10898-019-00741-w

On three soft rectangle packing problems with guillotine constraints

Quoc Trung Bui[1,2,3] · Thibaut Vidal[4] · Minh Hoàng Hà[5]

Received: 6 May 2018 / Accepted: 12 January 2019

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 $\mathcal{O}(n \log n)$, while the two others are NP-hard. We propose mixed integer linear programming 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 · Mixed integer linear 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 given areas $(a_1, \ldots, a_n)$,

✉ Minh Hoàng Hà
minhhoang.ha@vnu.edu.vn

Quoc Trung Bui
trungbui@daily-opt.com

Thibaut Vidal
vidalt@inf.puc-rio.br

1 Daily-Opt Joint Stock Company, Hanoi, Vietnam

2 FPT University, Hanoi, Vietnam

3 Risk division, Vietnam Technological and Commercial Joint Stock Bank, Hanoi, Vietnam

4 Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brazil

5 ORLab, University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam

Published online: 22 January 2019

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, and the largest aspect ratio, leading to three problems called COL-PERI-SUM, COL-PERI-MAX, and COL-ASPECT-RATIO respectively. Finally, the layout of the rectangles is subject to strict rules. As illustrated in Fig. 1, the rectangles should be delimited by *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 *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) is given by:

$$w(S_k) = \frac{\sum_{i \in S_k} a_i}{L_1}, \tag{1}$$

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:

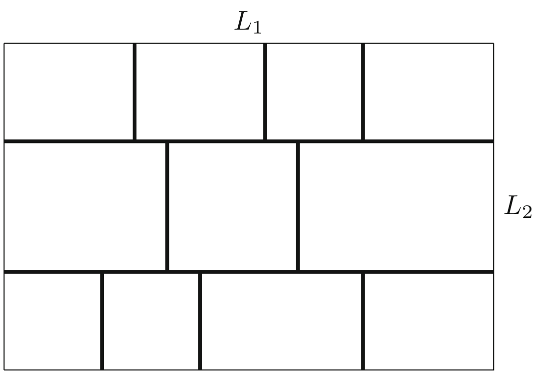
$$\begin{aligned} \text{COL-PERI-SUM: } \Phi_1 &= 2 \times \sum_{k=1}^{m} \sum_{i \in S_k} \left(w(S_k) + \frac{a_i}{w(S_k)} \right) \\ &= 2 \times \sum_{k=1}^{m} (|S_k| w(S_k) + L_1) \end{aligned} \tag{2}$$

$$\text{COL-PERI-MAX: } \Phi_2 = 2 \times \max_k \max_{i \in S_k} \left(w(S_k) + \frac{a_i}{w(S_k)} \right) \tag{3}$$

$$\text{COL-ASPECT-RATIO: } \Phi_3 = \max_k \max_{i \in S_k} \max \left(\frac{a_i}{w(S_k)^2}, \frac{w(S_k)^2}{a_i} \right). \tag{4}$$

The main contributions of the paper are as follows. First, we establish a connection between these three soft rectangle packing problems and the current Vietnamese land-allocation reform. We characterize their computational complexity, propose an efficient $\mathcal{O}(n \log n)$ algorithm for COL-PERI-SUM and demonstrate that the two other problems are NP-hard. Second, we introduce mixed integer linear programming (MILP) formulations for the NP-hard problems. For that purpose, the non-linear objective function of COL-ASPECT-RATIO is handled via a change of objective or binary search. Finally, we conduct experimental analyzes

Fig. 1 Partitioning the rectangular area by two-stage guillotine cuts—example solution



to determine the size limit of the instances which can be efficiently solved, and compare the solutions obtained with different objectives.

2 Applications and related work

2.1 Land reform in Vietnam

The agricultural land of Vietnam has been historically classified into several categories. 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 [28,29]. 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, where each plot has an average area of only ten square meters [21].

The land fragmentation turned out to be detrimental in the industrialized era. First of all, households cannot use machines to cultivate small plots, leading to a high production cost. Moreover, fragmented plots are costly to maintain, and the excessive number of tracks separating the plots causes a waste of agricultural land [14,17,24,25,29]. 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 in order to merge small plots into large fields and repartition these fields into larger plots for households. This reform has led to successful results in some provinces, as characterized by a significant increase in rice yield attaining 25% in Quang Nam province [7,29].

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 partition 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. Local authorities aim to minimize wasted land due to the creation of tracks, a goal which is captured by the Col-Peri-Sum objective. In contrast, farmers wish to have their plots as square as possible to facilitate cultivation. This goal can be expressed as a worst-case optimization to ensure a fair allocation, leading to the Col-Peri-Max and Col-Aspect-Ratio objectives. These objectives are not strongly conflicting, but they often lead to different solutions.

Land-consolidation strategies have been implemented in various other countries, e.g., in Germany [4–6], Turkey [8–10,16], Japan [1], Cyprus [11], China [18], and Brazil [15]. 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.

2.2 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. [2,3] 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 PERI-SUM and 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 [26] considered the general PERI-SUM, PERI-MAX and ASPECT-RATIO problems without guillotine constraints. The authors introduced an $\mathcal{O}(n \log n)$-time algorithm which produces a 1.25-approximate solution for PERI-SUM, a $2/\sqrt{3}$-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. [12] also designed an $2/\sqrt{3}$-approximate algorithm and a branch-and-cut algorithm for PERI-SUM.

Other close variants of PERI-SUM have been studied. Kong et al. [22,23] 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 [20,27,31]. Ibaraki and Nakamura [19] proposed a local search and mathematical programming algorithm to solve rectangular packing problems, where the shapes of the rectangles are adjustable within certain perimeter limits.

Finally, Beaumont et al. [3] 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 $\mathcal{O}(n^2 \log n)$ algorithm for COL-PERI-SUM and two approximation algorithms for COL-PERI-MAX. The complexity status of COL-PERI-MAX remains open. Moreover, COL-ASPECT-RATIO has not been studied up to this date. These methodological gaps are a strong motivation for additional research.

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

A polynomial-time algorithm in $\mathcal{O}(n^2 \log n)$ for COL-PERI-SUM was proposed in [3]. In this section, we introduce a simple algorithm in $\mathcal{O}(n \log n)$ 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 $\mathcal{O}(n)$ to the concave least-weight subsequence problem (CLWS), solvable to optimality in $\mathcal{O}(n)$ time [30].

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.

We first assume that the indices of the rectangles have been ordered in $\mathcal{O}(n \log n)$ by non-decreasing area: $a_1 \leq \cdots \leq a_n$. Then, we highlight a property of COL-PERI-SUM which allows us to focus the search on a smaller subset of solutions.

Theorem 1 *Consider a solution* $\mathbf{s}$ *of* COL-PERI-SUM *with cost* $\Phi_1(\mathbf{s})$*, 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* $\mathbf{s}'$ *obtained by swapping these two rectangles within their respective subsets is such that:*

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

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

$$\begin{aligned} \Delta &= \Phi_1(\mathbf{s}') - \Phi_1(\mathbf{s}) \\ &= \frac{2}{L_1}\left(|S_k|\left(a_j - a_i + \sum_{x \in S_k} a_x\right) + |S_l|\left(a_i - a_j + \sum_{x \in S_l} a_x\right) - |S_k| \sum_{x \in S_k} a_x - |S_l| \sum_{x \in S_l} a_x\right) \\ &= \frac{2}{L_1}\left(|S_k| - |S_l|\right)\left(a_j - a_i\right). \end{aligned}$$

□

This theorem defines some important features of the optimal solutions for 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 $j \in |S_l|$.

As a consequence, there exists an optimal solution $\mathbf{s}^* = \{S_1, \ldots, S_m\}$ 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:

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

where $w_{\mathrm{P}}(i, j)$ 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* $w_{\mathrm{P}}(i, j)$ *is concave.*

Proof For $0 \leq i_0 < i_1 < j_0 < j_1 \leq n$, one can directly verify that:

$$\begin{aligned}
\Delta' &= w(i_0, j_1) + w(i_1, j_0) - w(i_0, j_0) - w(i_1, j_1) \\
&= \frac{2}{L_1}\left((j_1 - i_0)\sum_{x=i_0+1}^{j_1} a_x + (j_0 - i_1)\sum_{x=i_1+1}^{j_0} a_x - (j_0 - i_0)\sum_{x=i_0+1}^{j_0} a_x - (j_1 - i_1)\sum_{x=i_1+1}^{j_1} a_x\right) \\
&= \frac{2}{L_1}\left((i_1 - i_0)\sum_{x=j_0+1}^{j_1} a_x + (j_1 - j_0)\sum_{x=i_0+1}^{i_1} a_x\right) > 0.
\end{aligned}$$

□

Therefore, after a prior ordering of the rectangles in $\mathcal{O}(n \log n)$, an optimal solution of Col-Peri-Sum can be found by solving an instance of CLWS, e.g., using the $\mathcal{O}(n)$ algorithm of Wilber [30]. Col-Peri-Sum can thus be solved in $\mathcal{O}(n \log n)$ in the general case, and in $\mathcal{O}(n)$ 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 proposed an efficient $\mathcal{O}(n \log n)$ algorithm for Col-Peri-Sum. In contrast, we will show 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-Φ and Col-Aspect-Ratio- Φ be the decision problems in which one must determine whether there exists a solution of value at most Φ for Col-Peri-Max, and Col-Aspect-Ratio, respectively. We will show that these two problems are NP-complete, by reduction from 2-Partition [13], hence establishing the NP-hardness of Col-Peri-Max and Col-Aspect-Ratio.

Theorem 3 Col-Peri-Max-Φ *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 $S_1 \cup S_2 = \{1, \ldots, n\}$, $S_1 \cap S_2 = \varnothing$ 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-Φ 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 $\sum_{x \in S_1} c_x = \sum_{x \in S_2} c_x$. Consider a solution of Col-Peri-Max-Φ 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 $\frac{L_2}{2} = c_{\text{MAX}}$. 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_{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_{MAX}. Hence, the soft rectangle of area c_{MAX}^2 is not arranged as a square, its perimeter exceeds $4 \times c_{\text{MAX}}$, and Col-Peri-Max-Φ is False. □

Theorem 4 Col-Aspect-Ratio- Φ *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- Φ 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 COL-ASPECT-RATIO- Φ is TRUE.

Now, assume that 2-PARTITION is FALSE. We distinguish three possible classes of solutions:

- Consider a solution of COL-ASPECT-RATIO with one layer. The rectangle of size $\frac{1}{M}$ has an aspect ratio of $4M$, which exceeds Φ.
- Consider a solution of 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 distinguished:
  - 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 $l_2 \leq 2 \times \frac{\frac{2}{M}}{2M+\frac{2}{M}+C}$. 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 $l_1 \geq \frac{c_i}{C+\frac{2}{M}} \times (M + \frac{1}{M} + \frac{C}{2})$ and $l_2 \leq 2 \times \frac{C+\frac{2}{M}}{2M+\frac{2}{M}+C}$. Thus,
  $$\Phi \geq \frac{c_i(M + \frac{1}{M} + \frac{C}{2})^2}{(C + \frac{2}{M})^2} > \frac{\min_{i=1}^{n} c_i \times M^2}{(C+1)^2} = M.$$
- Finally, consider a solution of COL-ASPECT-RATIO with two layers, where each layer contains exactly one rectangle of size M. Since there is no feasible solution of 2-PARTITION, the total areas of the layers are different (the smaller rectangles are too small to re-balance the sum). 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 Φ, and thus COL-ASPECT-RATIO- Φ is FALSE. □

5 Mixed integer linear programming models

Since COL-PERI-MAX and COL-ASPECT-RATIO are NP-hard, we propose MILP formulations for these problems. These formulations 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 Eqs. (5)–(17):

$$\min \quad \Phi_2 \tag{5}$$

$$\text{s.t.} \ \ 2(L_1 + L_2)(x_{ik} - 1) + 2\left(w_{ik} + \sum_{j=1}^{n} \frac{a_j x_{jk}}{L_1}\right) \le \Phi_2 i, \quad k \in \{1, \dots, n\} \tag{6}$$

$$\sum_{k=1}^{n} x_{ik} = 1 \quad i \in \{1, \dots, n\} \tag{7}$$

$$\sum_{i=1}^{n} x_{ik} \ge y_k \quad k \in \{1, \dots, n\} \tag{8}$$

$$x_{ik} \le y_k \quad i, k \in \{1, \dots, n\} \tag{9}$$

$$\sum_{i=1}^{n} w_{ik} = L_1 y_k \quad k \in \{1, \dots, n\} \tag{10}$$

$$w_{ik} \le L_1 x_{ik} i, \quad k \in \{1, \dots, n\} \tag{11}$$

$$a_i x_{ik} \le L_2 w_{ik} i, \quad k \in \{1, \dots, n\} \tag{12}$$

$$a_j w_{ik} - a_i w_{jk} \le a_j L_1 (2 - x_{ik} - x_{jk}) i, j, \quad k \in \{1, \dots, n\}, \quad i \ne j \tag{13}$$

$$a_i w_{jk} - a_j w_{ik} \le a_i L_1 (2 - x_{ik} - x_{jk}) \quad i, j, k \in \{1, \dots, n\}, \quad i \ne j \tag{14}$$

$$x_{ji} \in \{0, 1\} \quad i, j \in \{1, \dots, n\} \tag{15}$$

$$w_{ik} \ge 0 \quad i, k \in \{1, \dots, n\} \tag{16}$$

$$y_k \in \{0, 1\} \quad k \in \{1, \dots, n\} \tag{17}$$

$$\Phi_2 \ge 0 \tag{18}$$

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 contained 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. Constraints (11) and (12) impose that $(w_{ik} = 0) \Leftrightarrow (x_{ik} = 0)$. 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:

$$y_k \ge y_{k+1} \qquad k \in \{1, \dots, n-1\} \tag{19}$$

$$x_{ik} = 0 \qquad i \in \{1, \dots, n\}, k \in \{i+1, \dots, n\} \tag{20}$$

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 $k \in \{1, \dots, i\}$.

5.2 Formulation of COL-ASPECT-RATIO

The objective function Φ_3 is nonlinear, and we did not find a direct MILP 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- Φ can be formulated as a MILP. Solving this subproblem in a binary search allows us to solve the original optimization problem.

5.2.1 First approach—change of objective function

We introduce an alternative objective function Φ_4 for COL-ASPECT-RATIO, expressed as:

$$\Phi_4 = \max_{k} \max_{i \in S_k} \frac{|w(S_k) - \frac{a_i}{w_(S_k)}|}{\sqrt{a_i}}. \tag{21}$$

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,

$$\begin{aligned}
&\frac{\max(l_i, h_i)}{\min(l_i, h_i)} \geq \frac{\max(l_j, h_j)}{\min(l_j, h_j)} \\
\iff & \frac{l_i}{h_i} \geq \frac{l_j}{h_j} \\
\iff & \sqrt{\frac{l_i}{h_i}} \geq \sqrt{\frac{l_j}{h_j}} \\
\iff & \left(\sqrt{\frac{l_i}{h_i}} - \sqrt{\frac{l_j}{h_j}}\right)\left(1 + \frac{1}{\sqrt{\frac{l_i}{h_i}\frac{l_j}{h_j}}}\right) \geq 0 \\
\iff & \sqrt{\frac{l_i}{h_i}} - \sqrt{\frac{h_i}{l_i}} \geq \sqrt{\frac{l_j}{h_j}} - \sqrt{\frac{h_j}{l_j}} \\
\iff & \frac{l_i - h_i}{\sqrt{l_i h_i}} \geq \frac{l_j - h_j}{\sqrt{l_j h_j}} \\
\iff & \frac{|l_i - h_i|}{\sqrt{l_i h_i}} \geq \frac{|l_j - h_j|}{\sqrt{l_j h_j}}
\end{aligned}$$

□

Theorem 5 *Let s_3 and s_4 be two optimal solutions obtained with objectives Φ_3 and Φ_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 Φ_4 and Φ_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 $\Phi_4(s) = \frac{|l_i - h_i|}{\sqrt{l_i h_i}}$ and $\Phi_3(s) = \frac{\max(l_i, h_i)}{\min(l_i, h_i)}$.
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 Φ_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 Φ_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 Φ_3. A similar proof shows that s_3 is an optimal solution for objective Φ_4. □

Based on this change of objective function, Col-Aspect-Ratio can be formulated as:

$$\min \quad \Phi$$
$$\text{s.t.} \quad \text{Constraints (7)–(15)}$$
$$\delta_{ik} + L_1(1 - x_{ik}) \geq w_{ik} - \sum_{j=1}^{n} \frac{a_j x_{jk}}{L_1} i, k \in \{1, \ldots, n\} \tag{22}$$
$$\delta_{ik} + L_2(1 - x_{ik}) \geq -w_{ik} + \sum_{j=1}^{n} \frac{a_j x_{jk}}{L_1} i, k \in \{1, \ldots, n\} \tag{23}$$
$$\Phi \geq \frac{\delta_{ik}}{\sqrt{a_i}} i, k \in \{1, \ldots, n\} \tag{24}$$
$$\delta_i^k \geq 0 i, k \in \{1, \ldots, n\}. \tag{25}$$

Solving this formulation to optimality generates an optimal solution for Col-Aspect-Ratio. The value of this solution must be recomputed a-posteriori according to the original objective. The model uses n^2 additional continuous variables δ_{ik}, as well as a continuous variable Φ 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 COL-ASPECT-RATIO-Φ as a MILP. In this case, a maximum aspect ratio Φ is set as a constraint, and the goal is to find a feasible solution. The feasibility model can be written as follows:

$$\text{Constraints (7)–(15)}$$

$$L_1 h_k = \sum_{i=1}^{n} a_i x_{ik} k \in \{1, \ldots, n\} \tag{26}$$

$$w_{ik} \leq \Phi h_k i, k \in \{1, \ldots, n\} \tag{27}$$

$$h_k \leq \Phi w_{ik} i, k \in \{1, \ldots, n\} \tag{28}$$

$$h_k \in \mathbb{R} k \in \{1, \ldots, n\} \tag{29}$$

In this model, each variable h_k for $k \in \{1, \ldots, n\}$ stores the height of layer S_k, while Constraints (27)–(28) force the aspect ratio to be no higher than Φ. To solve the original optimization problem, we perform a binary search over Φ and solve COL-ASPECT-RATIO- Φ at each step. The starting interval is set to $[\Phi^{\text{LOW}}, \Phi^{\text{UP}}]$, where $\Phi^{\text{LOW}} = 1$, $\Phi^{\text{UP}} = \Phi_3(s_1)$, and s_1 is an optimal solution for COL-PERI-SUM found in $\mathcal{O}(n \log n)$ 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 10 GB 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. 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: $X \sim \mathcal{U}(1, 200)$.
- **Class MU**—The area of each item is sampled in a mixture of three uniform distributions: $X \sim \frac{1}{3}\left[\mathcal{U}(1, 10) + \mathcal{U}(11, 50) + \mathcal{U}(51, 150)\right]$.
- **Class MN**—The area of each item is sampled in a mixture of three normal distributions: $X \sim \frac{1}{3}\left[\mathcal{N}(5, 2) + \mathcal{N}(25, 10) + \mathcal{N}(125, 50)\right]$, but another sample is taken whenever the area is larger than 200.

Finally, the dimensions of the hard rectangle are generated as follows. 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, $A - L_1 L_2$ soft rectangles are randomly selected, and the area of each rectangle is reduced by one unit. After this procedure, 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.

6.1 Performance analysis

This section compares the CPU time needed to solve Col-Peri-Sum, Col-Peri-Max, and Col-Aspect-Ratio. As expected, the solution of 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 Col-Peri-Max as well as the reformulation and binary search approaches for Col-Aspect-Ratio. To speed up the solution methods, we always generate the optimal solution for Col-Peri-Sum and use it as an initial feasible solution.

Tables 1, 2 and 3 report, for each instance class and algorithm, the number of nodes in the search tree (**Nodes**), the CPU time in seconds (**Time**), as well as the best lower bound (**LB**) and upper bound (**UB**) found. For the reformulation-based approach for Col-Aspect-Ratio, columns $\mathbf{LB}_4$ and $\mathbf{UB}_4$ correspond to objective Φ_4, and the value of the primal solution for objective Φ_3 is indicated in column $\mathbf{UB}_3$. TL in column **Time** means that the CPU time limit of 3600 seconds has been attained. Finally, for the binary search approach for Col-Aspect-Ratio, we indicate the number of completed iterations in column $\mathbf{It}_{\mathbf{BS}}$.

As observed in these experiments, the proposed MILP 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 Col-Peri-Max and 40 rectangles for Col-Aspect-Ratio. Yet, the number of 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 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 Φ_4. When optimality is attained, this solution is optimal for Φ_3 due to Theorem 5. When an optimality gap remains, the primal solution obtained from the algorithm gives a valid upper bound for objective Φ_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 (**U**, **MU** and **MN**). We noted that two larger instances with 35 and 40 rectangles were solved to optimality for class **MU**, a phenomenon which did not happen for **U** and **MN**.

In general, the limitations of available mathematical programming algorithms are already visible when solving small instances (with 10 to 40 rectangles) of Col-Peri-Max and Col-Aspect-Ratio. Future progress on exact approaches for NP-hard problems may allow to solve larger instances in the future, but extensive 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, 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 $\mathcal{O}(n \log n)$, 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 for

Table 1 Class U—performance comparisons

Data		COL-PERI-MAX			COL-ASPECT-RATIO—reformulation						COL-ASPECT-RATIO—B. search			
#	Size	Nodes	Time	LB	UB	Nodes	Time	LB_4	UB_4	UB_3	It_{BS}	Time	LB	UB
p01	10	547	0.53	52.53	52.53	176	0.39	0.95	0.95	2.51	8	1.21	2.51	2.51
p02	10	27	0.20	55.17	55.17	68	0.22	1.72	1.72	4.75	10	1.38	4.75	4.75
p03	10	101	0.20	55.54	55.54	1	0.08	1.05	1.05	2.73	8	0.93	2.72	2.73
p04	15	5139k	TL	52.27	55.14	69	0.74	0.82	0.82	2.22	8	4.91	2.22	2.22
p05	15	38k	92.46	51.38	51.38	193	1.85	0.99	0.99	2.59	9	4.27	2.59	2.60
p06	15	8054k	TL	48.63	52.46	640k	479.41	1.98	1.98	5.76	10	560.23	5.75	5.76
p07	20	351k	TL	43.27	56.57	308	3.38	2.60	2.60	8.64	12	29.04	8.63	8.64
p08	20	2993k	TL	34.76	56.14	79k	202.12	0.47	0.47	1.59	7	174.54	1.59	1.59
p09	20	92	3.37	53.71	53.71	3k	12.31	1.46	1.46	3.86	9	35.94	3.85	3.86
p10	25	479k	TL	49.00	55.86	80k	851.98	0.74	0.74	2.07	8	679.89	2.06	2.07
p11	25	935k	TL	45.82	55.14	116	12.37	1.60	1.60	4.33	7	TL	4.29	4.37
p12	25	601k	TL	33.01	53.96	393k	TL	0.81	1.01	2.65	2	TL	2.62	4.25
p13	30	335k	TL	25.86	54.70	293k	TL	0.00	0.86	2.30	2	TL	2.20	3.41
p14	30	318k	TL	40.41	54.11	179k	TL	0.22	1.79	5.01	3	TL	1.00	7.22
p15	30	182k	TL	33.64	53.37	328k	TL	1.71	2.59	8.57	3	TL	1.00	17.06
p16	35	389k	TL	40.00	56.43	714k	TL	0.00	1.79	5.00	2	TL	1.00	5.34
p17	35	247k	TL	36.01	56.43	150k	TL	0.00	1.47	3.92	0	TL	1.00	8.34
p18	35	109k	TL	41.35	55.71	149k	TL	0.34	2.59	8.58	1	TL	1.00	28.70
p19	40	74k	TL	19.30	56.43	108k	TL	0.64	1.42	3.76	1	TL	1.00	5.86
p20	40	101k	TL	44.48	54.79	54k	TL	0.12	1.09	2.83	1	TL	1.93	2.86
p21	40	53k	TL	44.12	56.14	68k	TL	0.05	2.73	9.35	2	TL	1.00	21.33

Table 2 Class MU—performance comparisons

Data		COL-PERI-MAX				COL-ASPECT-RATIO—reformulation					COL-ASPECT-RATIO—B. search			
#	Size	Nodes	Time	LB	UB	Nodes	Time	LB_4	UB_4	UB_3	It_{BS}	Time	LB	UB
p01	10	179	0.31	55.29	55.29	17	0.24	1.39	1.39	3.65	9	1.03	3.64	3.65
p02	10	44	0.07	55.78	55.78	214	0.18	1.36	1.36	3.57	9	1.38	3.56	3.57
p03	10	4k	1.22	52.76	52.76	64	0.18	1.97	1.97	5.71	9	0.72	5.70	5.71
p04	15	1	0.49	56.17	56.17	720	1.67	2.20	2.20	6.67	11	12.05	6.66	6.67
p05	15	6938k	TL	43.43	51.23	294k	280.22	1.36	1.36	3.57	9	451.13	3.56	3.57
p06	15	185k	111.49	54.70	54.70	314	1.87	2.13	2.13	6.39	10	8.82	6.39	6.39
p07	20	1943k	TL	40.10	55.86	475k	1885.98	2.09	2.09	6.19	10	1450.33	6.19	6.19
p08	20	2223k	TL	28.26	53.81	1787k	TL	0.95	1.05	2.74	2	TL	2.69	4.38
p09	20	66k	285.98	55.86	55.86	142k	550.19	2.18	2.18	6.61	12	2355.92	6.61	6.61
p10	25	688k	TL	40.79	55.71	281k	TL	0.76	1.33	3.48	3	TL	2.98	3.96
p11	25	169k	TL	51.08	54.85	220k	TL	1.47	2.24	6.86	2	TL	1.00	12.49
p12	25	331k	TL	47.41	56.17	225k	TL	1.77	2.13	6.39	5	TL	5.72	6.66
p13	30	325k	TL	21.43	53.07	318k	TL	0.61	1.40	3.68	1	TL	1.00	7.02
p14	30	196k	TL	42.46	55.43	176k	TL	1.28	1.96	5.67	2	TL	3.69	6.38
p15	30	267k	TL	32.40	55.43	150k	TL	0.91	1.86	5.27	2	TL	1.00	7.10
p16	35	115k	TL	41.47	55.86	62k	TL	0.27	2.06	6.09	1	TL	4.94	8.89
p17	35	250k	TL	23.28	44.36	5k	297.25	1.39	1.39	3.67	9	471.57	3.67	3.67
p18	35	226k	TL	32.23	55.28	143k	TL	0.04	1.96	5.68	1	TL	1.00	7.20
p19	40	204k	TL	21.41	54.70	7k	181.11	1.48	1.48	3.92	4	TL	3.87	4.08
p20	40	290k	TL	29.09	56.00	72k	TL	0.11	1.53	4.11	3	TL	3.88	5.32
p21	40	154k	TL	15.76	52.76	75k	TL	0.00	1.30	3.39	0	TL	1.00	4.60

Table 3 Class MN—performance comparisons

Data		Col-Peri-Max				Col-Aspect-Ratio—reformulation					Col-Aspect-Ratio—B. search			
#	Size	Nodes	Time	LB	UB	Nodes	Time	LB_4	UB_4	UB_3	It_{BS}	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	1699.02	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	169k	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

Table 4 Optimal solutions for one objective evaluated according to the other objectives

Evaluated as	COL-PERI-SUM—Φ_1			COL-PERI-MAX—Φ_2			COL-ASPECT-RATIO—Φ_3		
Solved as	Φ_1	Φ_2	Φ_3	Φ_1	Φ_2	Φ_3	Φ_1	Φ_2	Φ_3
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

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 Φ_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 Φ_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 for 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 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, 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 proposed an $\mathcal{O}(n \log n)$ exact algorithm for COL-PERI-SUM. Then, we demonstrated that the two others problems are NP-hard, and proposed compact MILP formulations to solve them. In the case of 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 MILP formulations is currently practicable for problem instances involving 10 to 40 soft rectangles. For larger instances of COL-PERI-MAX and COL-ASPECT-RATIO, we also observed that the $\mathcal{O}(n \log n)$-time algorithm for COL-PERI-SUM produces good average results, allowing to use it as a simple and effective heuristic for these problems.

The research perspectives are numerous. The proposed 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 COL-PERI-MAX and COL-ASPECT-RATIO are *strongly* NP-hard remains an interesting open question.

Acknowledgements This research has been partly funded by the the National Council for Scientific and Technological Development (CNPQ–grant number 308498/2015-1) and FAPERJ in Brazil (grant number E-26/203.310/2016). This support is gratefully acknowledged.

References

1. Arimoto, Y.: Impact of land readjustment project on farmland use and structural adjustment: the case of Niigata, Japan. In: Agricultural and Applied Economics Association 2010 AAEA, CAES, WAEA Joint Annual Meeting, Denver, USA (2010)
2. Beaumont, O., Boudet, V., Rastello, F., Robert, Y.: Matrix multiplication on heterogeneous platforms. IEEE Trans. Parallel Distrib. Syst. **12**(10), 1033–1051 (2001)
3. Beaumont, O., Boudet, V., Rastello, F., Robert, Y.: Partitioning a square into rectangles: NP-completeness and approximation algorithms. Algorithmica **34**(3), 217–239 (2002)
4. Borgwardt, S., Brieden, A., Gritzmann, P.: Constrained minimum-k-star clustering and its application to the consolidation of farmland. Oper. Res. **11**(1), 1–17 (2011)
5. Borgwardt, S., Brieden, A., Gritzmann, P.: Geometric clustering for the consolidation of farmland and woodland. Math. Intell **36**(2), 37–44 (2014)
6. Brieden, A., Gritzmann, P.: A quadratic optimization model for the consolidation of farmland by means of lend-lease agreements. In: Ahr, D., Fahrion, R., Oswald, M., Reinelt, G. (eds.) Operations Research Proceedings 2003, pp. 324–331. Springer, Berlin, Heidelberg (2004)

7. Bui, Q.T., Pham, Q.D., Deville, Y.: Solving the agricultural land allocation problem by constraint-based local search. In: Schulte, C. (ed.) Principles and Practice of Constraint Programming. Lecture Notes in Computer Science, vol. 8124, pp. 749–757. Springer, Berlin, Heidelberg (2013)
8. Cay, T., Uyan, M.: Evaluation of reallocation criteria in land consolidation studies using the analytic hierarchy process (AHP). Land Use Policy **30**(1), 541–548 (2013)
9. Cay, T., Ayten, T., Iscan, F.: An investigation of reallocation model based on interview in land consolidation. In: Proceeding of the 23rd FIG Congress, Shaping the Change, Munich, Germany (2006)
10. Cay, T., Ayten, T., Iscan, F.: Effects of different land reallocation models on the success of land consolidation projects: social and economic approaches. Land Use Policy **27**(2), 262–269 (2010)
11. Demetriou, D., See, L., Stillwell, J.: A spatial genetic algorithm for automating land partitioning. Int. J. Geogr. Inf. Sci. **27**, 2391–2409 (2013)
12. Fügenschuh, A., Junosza-Szaniawski, K., Lonc, Z.: Exact and approximation algorithms for a soft rectangle packing problem. Optim. J. Math. Program. Oper. Res. **63**(11), 1637–1663 (2014)
13. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1990)
14. General Statistical Office: Statistical Yearbook 2004. Statistical Publishing House, Hanoi (2004)
15. Gliesch, A., Ritt, M., Moreira, MCO.: A genetic algorithm for fair land allocation. In: Proceedings of the Genetic and Evolutionary Computation Conference, ACM, New York, NY, USA, GECCO '17, pp. 793–800 (2017)
16. Hakli, H., Uuz, H.: A novel approach for automated land partitioning using genetic algorithm. Expert Syst. Appl. **82**(1), 10–18 (2017)
17. Heltberg, R.: Rural market imperfections and the farm size-productivity relationship: evidence from Pakistan. World Dev. **26**, 1807–1826 (1998)
18. Huang, Q., Li, M., Chen, Z., Li, F.: Land consolidation: an approach for sustainable development in rural China. AMBIO **40**(1), 93–95 (2011)
19. Ibaraki, T., Nakamura, K.: Packing problems with soft rectangles. In: Almeida, F., Blesa Aguilera, M.J., Blum, C., Moreno Vega, J.M., Pérez Pérez, M., Roli, A., Sampels, M. (eds.) Hybrid Metaheuristics, pp. 13–27. Springer, Berlin, Heidelberg (2006)
20. Ji, P., He, K., Jin, Y., Lan, H., Li, C.: An iterative merging algorithm for soft rectangle packing and its extension for application of fixed-outline floorplanning of soft modules. Comput. Oper. Res. **86**, 110–123 (2017)
21. J. Rural Econ. (2008) Agricultural mechanization: When does it go over the start? http://ipsard.gov.vn/mobile/tID2264_Co-gioi-hoa-nong-nghiep-Khi-nao-qua-buoc-khoi-dong-.html. Accessed 01 May 2018
22. Kong, T.Y., Mount, D.M., Werman, M.: The decomposition of a square into rectangles of minimal perimeter. Discrete Appl. Math. **16**(3), 239–243 (1987)
23. Kong, T.Y., Mount, D.M., Werman, M.: The decomposition of a rectangle into rectangles of minimal perimeter. SIAM J. Comput. **17**(6), 1215–1231 (1988)
24. Lam, M.L.: Land fragmentation-a constraint for Vietnam agriculture. Vietnam Socio Econ. Dev. **26**, 73–80 (2001)
25. March, S.P., MacAulay, T.G.: Farm size and land use change in Vietnam following land reforms. In: 47th Annual Conference of the Australian Agricultural and Resource Economics Society, Fremantle, Australia (2006)
26. Nagamochi, H., Abe, Y.: An approximation algorithm for dissecting a rectangle into rectangles with specified areas. Discrete Appl. Math. **155**(4), 523–537 (2007)
27. Paes, F., Pessoa, A., Vidal, T.: A hybrid genetic algorithm with decomposition phases for the unequal area facility layout problem. Eur. J. Oper. Res. **256**(3), 742–756 (2017)
28. Pham, V.H., MacAulay, G.T., Marsh, S.P.: The economics of land fragmentation in the north of Vietnam. Aust. J. Agric. Resour. Econ. **51**(2), 195–211 (2007)
29. Sundqvist, P., Anderson, L.: A study of the impacts of land fragmentation on agricultural productivity in Northern Vietnam. Bachelor's thesis, Uppsala University, Sweden (2006)
30. Wilber, R.: The concave least-weight subsequence problem revisited. J. Algorithms **9**(3), 418–425 (1988)
31. Young, F.Y., Chu, C.C.N., Luk, W.S., Wong, Y.C.: Handling soft modules in general nonslicing floorplan using lagrangian relaxation. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **20**(5), 687–692 (2001)