Discrete Applied Mathematics 187 (2015) 120–129

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Minimal proper non-IRUP instances of the one-dimensional cutting stock problem

Vadim M. Kartak [a,b,*], Artem V. Ripatti [a], Guntram Scheithauer [c], Sascha Kurz [d]

[a] Bashkir State Pedagogical University named after M. Akmullah, Oktyabrskoy Revolutsii st. 3a, 450000, Ufa, Russia
[b] Ural Federal University named after the first President of Russia B.N. Yeltsin, Mira st. 19, 620002, Ekaterinburg, Russia
[c] Technische Universität Dresden, Mommsenstrasse 9, 01069, Dresden, Germany
[d] Universität Bayreuth, Universitätsstrasse 30, 95440 Bayreuth, Germany

ARTICLE INFO

Article history:
Received 29 May 2014
Received in revised form 12 February 2015
Accepted 16 February 2015
Available online 16 March 2015

Keywords:
Cutting stock problem
Bin packing problem
Integer round-up property
Equivalence of instances
Weighted simple games

ABSTRACT

We consider the well-known one dimensional cutting stock problem (1CSP). Based on the pattern structure of the classical ILP formulation of Gilmore and Gomory, we can decompose the infinite set of 1CSP instances, with a fixed number n of demanded pieces, into a finite number of equivalence classes. We show up a strong relation to weighted simple games. Studying the integer round-up property (IRUP) we use the proper LP relaxation of the Gilmore and Gomory model that allows us to consider the 1CSP as the bin packing problem (BPP). We computationally show that all 1CSP instances with $n \leq 9$ have the proper IRUP, while we give examples of proper non-IRUP instances with $n = 10$ and proper gap 1. Proper gaps larger than 1 occur for $n \geq 11$. The largest known proper gap is raised from 1.003 to 1.0625. The used algorithmic approaches are based on exhaustive enumeration and integer linear programming. Additionally we give some theoretical bounds showing that all 1CSP instances with some specific parameters have the proper IRUP.

1. Introduction

An integer linear programming (ILP) minimization problem has the integer round-up property (IRUP) if, for every instance, rounding up the optimal value of its linear programming (LP) relaxation yields the optimal value of the integer problem. This notation was introduced by Baum and Trotter [1]. Here we study the one-dimensional cutting stock problem (1CSP) with respect to the IRUP. The classical ILP formulation for the cutting stock problem by Gilmore and Gomory is based on so-called cutting patterns [8]. Using this formulation, Marcotte [15] has shown that certain subclasses of cutting stock problems have the IRUP, while she later showed that there are instances of the 1CSP having a gap of exactly 1 [16]. The first example with gap larger than 1 was given in [6]. Subsequently, the gap was increased to $\frac{6}{5}$ [18,20–22], i.e., no example with a gap of at least 2 is currently known. Indeed, the authors of [23] have conjectured that the gap is always below 2—called the modified integer round-up property (MIRUP), which is one of the most important open theoretical issues about the 1CSP, see also [5]. Practical experience shows that the typical gap is rather small [24]. An algorithm for verifying whether an 1CSP instance has the IRUP or not, is presented in [10].

Dropping some cutting patterns from the ILP formulation of [8], the authors of [17] introduced the *proper relaxation* of the 1CSP, whose optimal value is at least as large as the LP relaxation bound. For an instance of the 1CSP, we call the difference of the optimal value and the optimal value of the corresponding proper relaxation the *proper gap*. If the proper gap is at

* Corresponding author at: Bashkir State Pedagogical University named after M. Akmullah, Oktyabrskoy Revolutsii st. 3a, 450000, Ufa, Russia.
E-mail addresses: kvmail@mail.ru (V.M. Kartak), ripatti@inbox.ru (A.V. Ripatti), guntram.scheithauer@tu-dresden.de (G. Scheithauer), sascha.kurz@uni-bayreuth.de (S. Kurz).

http://dx.doi.org/10.1016/j.dam.2015.02.020
0166-218X/

least 1, we speak of a proper non-IRUP instance. We remark that the proper LP relaxation of the 1CSP is equivalent to the LP relaxation of the bin packing problem (BPP) considered in [3,4] (i.e. the proper gap for the 1CSP is the gap for the BPP). The currently largest known proper gap of the 1CSP is 1.003 [3,4].

Our aim is to find insights in the structural behavior of the ILP formulation of the 1CSP and its two relaxations.

The paper is organized as follows. In Section 2 we introduce the basic notation. The concept of partitioning the infinite set of 1CSP instances with fixed demand into a finite set of equivalence classes is described in Section 3. The relation to the discrete structure of weighted simple games, from cooperative game theory, is the topic of Section 3.1. In Section 4 we develop an exhaustive enumeration algorithm for the generation of all equivalence classes of 1CSP instances. We proceed with some theoretical bounds on the proper gap of 1CSP instances in Section 5, which are used to reduce the search space in order to find instances with maximal gap. Based on the ILP approaches in Section 6 we present computational results in Section 7. Finally, we draw a conclusion in Section 8.

2. Basic notation

We assume that one-dimensional material objects like, e.g. paper reels or wooden rods, of a given length $L \in \mathbb{R}_{>0}$ are cut into smaller pieces of lengths $l_1, \ldots, l_m \in \mathbb{R}_{>0}$ in order to fulfill the order demands $b_1, \ldots, b_m \in \mathbb{Z}_{>0}$. The question for the minimum needed total amount of stock material or, equivalently, the minimization of waste, is the famous 1CSP. Using the abbreviations $l = (l_1, \ldots, l_m)^T$ and $b = (b_1, \ldots, b_m)^T$ we denote an instance of the 1CSP by $E = (m, L, l, b)$.

The cutting patterns, mentioned in the introduction, are formalized as vectors $a = (a_1, \ldots, a_m)^\top \in \mathbb{Z}^m_{\geq 0}$. We say, a *pattern* a of E is *feasible* if $l^\top a \leq L$. By $P^f(E) := \left\{a \,:\, l^\top a \leq L,\ a \in \mathbb{Z}^m_{\geq 0}\right\}$ we denote the set of all feasible patterns. Additionally, we call a pattern *proper* if it is feasible and $a_i \leq b_i$ for all $1 \leq i \leq m$. By $P^p(E) := \left\{a \in P^f(E) \,:\, a_i \leq b_i,\ 1 \leq i \leq m\right\}$ we denote the set of all proper patterns.

Given a set of patterns $P = \{a^1, \ldots, a^r\}$ (of E), let $A(P)$ denote the $(m \times r)$-matrix whose columns are given by the vectors a^i. With this we can define

$$z_D(P, E) := \sum_{i=1}^{r} x_i \to \min \quad \text{subject to} \quad A(P)x = b,\ x \in \mathbb{Z}^r_{\geq 0} \quad \text{and}$$

$$z_C(P, E) := \sum_{i=1}^{r} x_i \to \min \quad \text{subject to} \quad A(P)x = b,\ x \in \mathbb{R}^r_{\geq 0}.$$

Choosing $P = P^f(E)$ we obtain the mentioned ILP formulation of Gilmore and Gomory for the 1CSP and its continuous (or LP) relaxation. As abbreviations we use $z^f_D(E) := z_D(P^f(E), E)$, $z^f_C(E) := z_C(P^f(E), E)$, and $\Delta(E) := z^f_D(E) - z^f_C(E)$, where the latter is called the *gap of instance* E. So, an instance E has the *IRUP* if $\Delta(E) < 1$. Otherwise, E is called a *non-IRUP instance*. Moreover, E is a *MIRUP instance* if $\Delta(E) < 2$.

Choosing $P = P^p(E)$ we obtain the proper relaxation with optimal value $z^p_C(E) := z_C(P^p(E), E)$. Since $z_D(P^p(E), E) = z_D(P^f(E), E)$, we call $\Delta_p(E) := z^f_D(E) - z^p_C(E)$ the *proper gap of instance* E. Similarly, an instance E is a *proper IRUP instance* if $\Delta_p(E) < 1$ and a *proper non-IRUP instance* otherwise.

Due to $\Delta_p(E) \leq \Delta(E)$, proper non-IRUP instances are non-IRUP instances too. The converse is not true as shown by the example $E = \left(3, 30, (2, 3, 5)^\top, (1, 2, 4)^\top\right)$ with $\Delta(E) = 31/30$ and $\Delta_p(E) = 4/5$.

3. Equivalence of 1CSP instances

Given an 1CSP instance $E = (m, L, l, b)$ with a total demand of $n = \sum_{i=1}^{m} b_i$ pieces, we can easily transform it into an instance $\bar{E} = (n, L, l', b')$ with $b'_i = 1$ for all $1 \leq i \leq n$ by taking b_j copies of length l_j for each $1 \leq j \leq m$. It is easy to verify that this transformation has no effect with respect to the optimal values of the stated ILPs and their relaxations, i.e., we have $z^f_D(E) = z^f_D(\bar{E})$, $z^f_C(E) = z^f_C(\bar{E})$, and $z^p_C(E) = z^p_C(\bar{E})$. Thus in view of our investigations, in the following we assume $b_i = 1$ for all i and abbreviate such 1CSP instances by $E = (n, L, l)$. In particular, now we have $P^p(E) \in \mathbb{B}^n$, where $\mathbb{B} = \{0, 1\}$. Without loss of generality, we assume $0 < l_1 \leq \cdots \leq l_n \leq L$ in the following.

We remark that, using this modification, the 1CSP becomes equivalent to the Bin Packing Problem (BPP), where n items of size l_i have to be packed into as few as possible identical bins of capacity L. So our results also hold for the BPP and indeed some part of the related literature considers the BPP instead of the 1CSP. The continuous relaxation of the BPP is also known as the Fractional Bin Packing Problem, cf. [4]. It corresponds to the proper relaxation of the 1CSP.

Since the set partitioning formulation by Gilmore and Gomory and its proper relaxation actually do not depend directly on the parameters L and l_i, $1 \leq i \leq n$, we partition the set of 1CSP instances with a fixed number n of demanded pieces into equivalence classes according to their set of proper patterns.

Definition 1. The 1CSP instances E and $\bar{E}$ are pattern-equivalent if $P^p(E) = P^p(\bar{E})$.

Since $z^p_C(E) = z^p_C(\bar{E})$, $z^f_D(E) = z^f_D(\bar{E})$, and, consequently, $\Delta_p(E) = \Delta_p(\bar{E})$ for pattern-equivalent instances E and $\bar{E}$, we can restrict ourselves onto the set of equivalence classes

$$\mathbb{P}^p_n := \left\{P^p(E) \,:\, E = (n, L, l),\ L \in \mathbb{R}_{>0},\ l \in \mathbb{R}^n_{>0}\right\}.$$

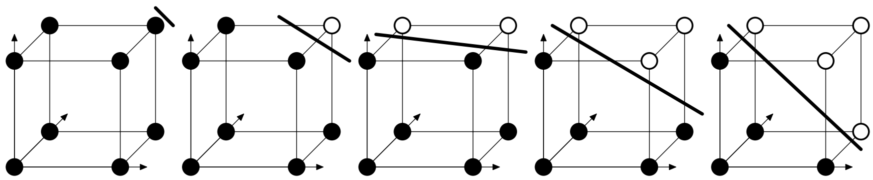

Fig. 1. All equivalence classes of $\mathbb{P}_3^p$.

We remark that the authors of [19] have introduced a finer equivalence relation, called full pattern-equivalence, by demanding $P^f(E) = P^f(\widehat{E})$. This is needed to, additionally, guarantee that $z_C^f(E) = z_C^f(\widehat{E})$ and $\Delta(E) = \Delta(\widehat{E})$. For example, the instances $E = (6, 30, (6, 6, 10, 10, 11, 15)^\top)$ and $\widehat{E} = (6, 10000, (2000, 2000, 3001, 3250, 3750, 5000)^\top)$ are proper pattern-equivalent but not full pattern-equivalent, because $P^f(\widehat{E})$ contains pattern $(0, 0, 2, 0, 1, 0)$ but $P^f(E)$ does not.

Obviously, $\left|\mathbb{P}_n^p\right| \le 2^{|\mathbb{B}^n|} = 2^{2^n}$ is finite, but not all subsets of $\mathbb{B}^n$ can be attained as proper patterns of an 1CSP instance.

Lemma 1. *Given $P \subseteq \mathbb{B}^n$, an 1CSP instance $E = (n, L, l)$ with $P^p(E) = P$ exists iff the following system of linear inequalities contains a solution:*

$$\begin{aligned} &1 \le l_1 \le \cdots \le l_n \le L, && (1)\\ &\sum_{i=1}^n l_i a_i \le L \quad \forall a \in P, \\ &\sum_{i=1}^n l_i a_i \ge L + 1 \quad \forall a \in \mathbb{B}^n \backslash P, \\ &l_1, l_2, \dots, l_n, L \in \mathbb{R}_{\ge 0}. \end{aligned}$$

Proof. Let $E = (n, L, l)$ be an 1CSP instance with $P = P^p(E)$. Due to definition we have $0 < l_1 \le \cdots \le l_n \le L$. By multiplying the l_i and L with a suitable positive factor, we can ensure $l_1 \ge 1$. Similarly, we have $\sum_{i=1}^n l_i a_i \le L$ for all $a \in P$ and $\sum_{i=1}^n l_i a_i > L$ for all $a \in \mathbb{B}^n \backslash P$ by definition, so that multiplying the variables with a suitable positive factor ensures that all constraints are satisfied.

If L and l satisfy the inequalities (1), then $E = (n, L, l)$ is an example of the demanded instance. □

We remark that we can additionally require that L and all l_i are positive integers, and indeed we will use only integers in our subsequent examples of 1CSP instances.

The parameters l_i, $1 \le i \le n$, and L of Lemma 1 have the following nice geometric interpretation. The hyperplane defined by $\sum_{i=1}^n l_i x_i = L$ perfectly separates the set of proper patterns $P^p(E)$ and the set of non-proper patterns $\mathbb{B}^n \backslash P^p(E)$ within the unit-hypercube. In Fig. 1 we have depicted all five equivalence classes for $n = 3$, where the proper patterns are marked by filled black circles.

So a first simple and finite algorithm to determine the maximum $\Delta_p(E)$ for a given number n of demanded pieces, is to loop over all equivalence classes in $\mathbb{P}_n^p$ and to compute the respective $\Delta_p(E)$. Of course this is possible for rather small n only. It may be of different interest to explicitly construct a complete system of representatives of $\mathbb{P}_n^p$. So, we present an enumeration algorithm in Section 4 and then we proceed with ILP approaches in Section 6. Prior to that we relate our discrete structures with another stream of literature in the context of cooperative game theory.

3.1. Relation of 1CSP instances to weighted simple games

In cooperative game theory a *simple game* on n voters is defined as a mapping $v : \mathbb{B}^n \to \mathbb{B}$ satisfying $v(\mathbf{0}) = 0$, $v(\mathbf{1}) = 1$, and $v(a) \le v(b)$ for all $a, b \in \mathbb{B}^n$ with $a \le b$, i.e., $a_i \le b_i$ for all $1 \le i \le n$ (cf. [25]). A vector[1] $a \in \mathbb{B}^n$ with $v(a) = 1$ is called *winning* and *losing* otherwise. Each simple game is uniquely characterized by either its set of winning or its set of losing vectors. A simple game v is called *weighted* if there exist weights $w \in \mathbb{R}_{\ge 0}^n$ and a quota $q \in \mathbb{R}_{>0}$ such that $v(a) = 1$ iff $a^\top w \ge q$. Without loss of generality, we can assume that the quota and the weights are positive integers with $1 \le w_1 \le \cdots \le w_n \le q$ and $q \ge 2$.

Given a weighted simple game v represented by weights $w \in \mathbb{Z}_{>0}$ and a quota $q \in \mathbb{Z}_{\ge 2}$, we can set $L = q - 1$ and $l_i = w_i$ for all $1 \le i \le n$. If additionally all unit-vectors are losing in v, then we have $l_i = w_i \le q - 1 = L$, i.e., $E = (n, L, l)$ is an 1CSP instance, where the losing vectors correspond to the proper patterns.

For the other direction let $E = (n, L, l)$ be an 1CSP instance with $l \in \mathbb{Z}_{>0}^n$ and $L \in \mathbb{Z}_{>0}$. If additionally the all-one vector $\mathbf{1}$ is a non-feasible pattern, then setting $q = L + 1$ and $w_i = l_i$ for all $1 \le i \le n$ yields a weighted simple game v.

[1] Mostly one speaks of subsets $S \subseteq \{1, \dots, n\}$, called coalitions, in the corresponding literature. The vectors we use here correspond to the incidence vectors of those sets.

4. Enumeration of all pattern-equivalent classes of the 1CSP with fixed *n*

The equivalence class of an 1CSP instance E is uniquely described by its set $P = P^p(E) \subseteq \mathbb{B}^n$ of proper patterns. In order to loop over all equivalence classes, we have to enumerate all possible choices for P and subsequently decide which pattern is a proper one and which is not. We observe that, in Lemma 1, infeasibility with respect to inequalities (1) can already be recognized by only a proper subset of patterns.

Lemma 2. *Given two disjoint subsets $P_{\le}$ and $P_{>}$ of $\mathbb{B}^n$. If*

$$
\begin{aligned}
&1 \le l_1 \le \cdots \le l_n \le L, \\
&\sum_{i=1}^{n} l_i a_i \le L \quad \forall a \in P_{\le}, \\
&\sum_{i=1}^{n} l_i a_i \ge L + 1 \quad \forall a \in P_{>}, \\
&l_1, l_2, \ldots, l_n, L \in \mathbb{R}_{\ge 0}
\end{aligned}
\tag{2}
$$

does not have a solution, then there cannot exist an 1CSP instance $E = (n, L, l)$ with $P_{\le} \subseteq P^p(E) \subseteq \mathbb{B}^n \backslash P_{>}$.

Next we observe that some inequalities of (1) and (2) may be dominated by others. For $a \le b$, i.e, $a_i \le b_i$ for all $1 \le i \le n$, we clearly have $l^\top a \le l^\top b$ due to $l \ge 0$. Using the given ordering $l_1 \le \cdots \le l_n$, we can even uncover more dominated inequalities. To this end we introduce the following binary relation.

Definition 2. For $a, b \in \mathbb{B}^n$ we write $a \preceq b$ iff $\sum_{i=j}^{n} a_i \le \sum_{i=j}^{n} b_i$ for all $1 \le j \le n$.

We say that a is *dominated* by b. In the context of simple games the relation $\preceq$, using the reverse ordering of coordinates, is used to define the class of so-called complete simple games, which is a subclass of weighted simple games, see [9,25]. So the following results are well known in a different context and we mention only the facts that we are explicitly using in the paper.

Lemma 3. *Let $a, b \in \mathbb{B}^n$ with $a \preceq b$. For $l_1 \le \cdots \le l_n$ we have $l^\top a \le l^\top b$.*

Proof. Setting $l_j = \sum_{i=1}^{j} k_i$, the $k_i \ge 0$ are uniquely defined and we have

$$
l^\top a = \sum_{j=1}^{n} \left(k_j \cdot \sum_{i=j}^{n} a_i \right) \le \sum_{j=1}^{n} \left(k_j \cdot \sum_{i=j}^{n} b_i \right) = l^\top b. \quad \square
$$

Corollary 1. *Let $\mathbf{0} \le a \preceq b \le \mathbf{1}$. If $b \in P^p(E)$, then $a \in P^p(E)$. If $a \notin P^p(E)$, then $b \notin P^p(E)$.*

Observation 1. $\{a \in \mathbb{B}^n : \|a\|_1 \le 1\} \subseteq P^p(E)$ *for all 1CSP instances $E = (n, L, l)$ due to $l_i \le L$ for all $1 \le i \le n$.*

With those ingredients we can state the following enumeration algorithm.

MainProcedure()
$P_{\le} \leftarrow \{a \in \mathbb{B}^n : \|a\|_1 = 1\}$
$P_{>} \leftarrow \emptyset$
$P_u \leftarrow \mathbb{B}^n \backslash \{a \in \mathbb{B}^n : \|a\|_1 \le 1\}$
RecursiveProcedure($P_{\le}$, $P_{>}$, P_u)

RecursiveProcedure($P_{\le}$,$P_{>}$,P_u)
if system (2) has no solution for $P_{\le}$ and $P_{>}$
    return
if $P_u = \emptyset$ $\quad\triangleright$ we have found a new equivalence class
    save $\left\{a \in \mathbb{B}^n : \exists b \in P_{\le} : a \preceq b\right\}$
    return
choose some pattern $a \in P_u$ $\quad\triangleright$ no matter which one
$P'_{\le} \leftarrow P_{\le}$
remove all patterns from $P'_{\le}$ which are dominated by a
$P'_u \leftarrow P_u \backslash \{b \in \mathbb{B}^n : b \preceq a\}$
RecursiveProcedure($P'_{\le} \cup \{a\}$, $P_{>}$, P'_u)
$P'_{>} \leftarrow P_{>}$
remove all patterns from $P'_{>}$ which dominate a
$P'_u \leftarrow P_u \backslash \{b \in \mathbb{B}^n : a \preceq b\}$
RecursiveProcedure($P_{\le}$, $P'_{>} \cup \{a\}$, P'_u)

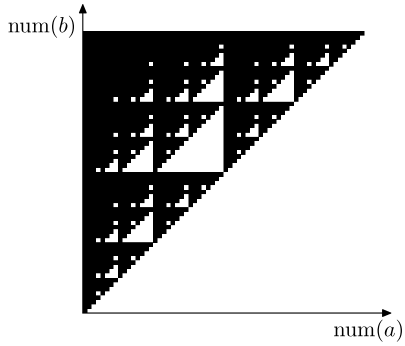

Fig. 2. Illustration of the dominance relation.

Here $P_{\leq}$ denotes the set of patterns which have been classified by the algorithm to be feasible. Similarly, $P_{>}$ denotes the set of the patterns which have to be non-feasible. Since all patterns with only a single element have to be feasible by definition, we can initialize $P_{\leq}$ as done in the `MainProcedure`.

Obviously, a pattern a is feasible if there exists a pattern $b \in P_{\leq}$ with $a \preceq b$ since b is feasible and proper. Similarly, pattern a is non-feasible, if there exists a pattern $b \in P_{>}$ with $b \preceq a$ since b is non-feasible. We remark that all unclassified patterns are pooled in the set P_u.

In order to save computation time within the check of the inequality system (2), we try to remove as many patterns as possible from $P_{\leq}$ and $P_{>}$ in lines 8 and 12 of `RecursiveProcedure`. Since $\mathbb{B}^n$ is a partially ordered set under $\preceq$, the constructed sets $P_{\leq}$ and $P_{>}$ are indeed minimal in every iteration. For $n = 9$ this approach reduces the computation time, due to the decreased number of inequalities in Lemma 2, by a factor of roughly 50.

The dominance relation $\preceq$ can be checked in $O(n)$. Since those comparisons occur quite often it is beneficial to compute and store them once for all pairs of patterns. Some comparisons can additionally be avoided by using:

Observation 2. *For* $\mathbf{0} \leq a \preceq b \leq \mathbf{1}$ *we have* $\mathrm{num}(a) \leq \mathrm{num}(b)$, *where* $\mathrm{num}(a) := \sum_{i=1}^{n} 2^{i-1} a_i$.

The converse is generally not true, i.e., $\mathrm{num}(a) \leq \mathrm{num}(b)$ implies either $a \preceq b$ or a and b are incomparable. In Fig. 2 we have depicted the dominance relation where the patterns are ordered by the num() function. Black squares represent the cases $a \preceq b$; white ones the cases $a \not\preceq b$.

5. Bounds for $\Delta_p(E)$

Obviously, we have $0 \leq z_C^p(E) \leq z_D^f(E) \leq n$ for each 1CSP instance $E = (n, L, l)$. The cases with $z_D^f(E) = 1$ can be completely classified:

Lemma 4. *For an* 1CSP *instance* $E = (n, L, l)$ *we have*

$$z_D^f(E) = 1 \iff \mathbf{1} \in P^p(E) \iff P^p(E) = \mathbb{B}^n \iff \sum_{i=1}^{n} l_i \leq L.$$

Corollary 2. *If* $z_D^f(E) = 1$, *then we have* $z_C^p(E) = 1$ *and* $\Delta_p(E) = 0$.

Also the cases where $z_D^f(E) = 2$ can be characterized completely:

Lemma 5. *We have* $z_D^p(E) > 2$ *if and only if* $\{a, \mathbf{1} - a\} \not\subseteq P^p(E)$ *for all* $a \in \mathbb{B}^n$.

Proof. Choosing $a = \mathbf{1}$ we conclude $a \notin P^p(E)$, since $\mathbf{0} \in P^p(E)$. Thus we can assume $z_D^f(E) > 1$. We have $z_D^f(E) = 2$ iff there exist feasible patterns $a, b \in P^p(E)$ with $a + b = \mathbf{1}$. Thus $b = \mathbf{1} - a \in P^p(E)$. $\square$

We remark that simple games, where not both coalition vectors a and $\mathbf{1} - a$ can be losing, are called *proper*.

The optimal solution of the proper relaxation is given by non-negative real multipliers γ_a satisfying

$$\sum_{a \in P^p(E)} \gamma_a \cdot a = \mathbf{1} \quad \text{and} \quad z_C^p(E) = \sum_{a \in P^p(E)} \gamma_a. \tag{3}$$

Lemma 6. $z_C^p(E) \geq 1$ *for any instance* E *of* 1CSP.

Table 1
Number of equivalence classes for $n = 8$ and a given $z_D^p(E)$-value.

$z_D^p(E)$	1	2	3	4	5	6	7	8
#	1	1 363 847	1 277 944	56 895	1992	103	8	1

Proof. From (3) we conclude

$$n \cdot \sum_{a \in P^p(E)} \gamma_a = \sum_{a \in P^p(E)} \gamma_a \cdot n \geq \sum_{a \in P^p(E)} \gamma_a \cdot \|a\|_1 = \|\mathbf{1}\|_1 = n. \quad \square$$

The above proof can be slightly tightened if $\mathbf{1} \notin P^p(E)$.

Lemma 7. *If $z_D^f(E) = 2$, then $z_C^p(E) \geq \frac{n}{n-1}$ and $\Delta_p(E) \leq \frac{n-2}{n-1}$.*

Proof. Since $z_D^f(E) \neq 1$ we have $\mathbf{1} \notin P^p(E)$ so that $\|a\|_1 \leq n-1$ for all patterns $a \in P^p(E)$. Combining this with (3) yields

$$(n-1) \cdot \sum_{a \in P^p(E)} \gamma_a = \sum_{a \in P^p(E)} \gamma_a \cdot (n-1) \geq \sum_{a \in P^p(E)} \gamma_a \cdot \|a\|_1 = \|\mathbf{1}\|_1 = n.$$

Thus we have $z_C^p(E) \geq \frac{n}{n-1}$ and $\Delta_p(E) = z_D^f(E) - z_C^p(E) \leq \frac{n-2}{n-1}$. $\square$

Lemma 8. *If $z_D^f(E) > 2$, then $z_C^p(E) > 2$ and therefore $\Delta_p(E) < z_D^f(E) - 2$.*

Proof. Without loss of generality, we assume that the parameters L and l_i, $1 \leq i \leq n$, of $E = (n, L, l)$ are integers and that there exists a feasible pattern $a \in P^p(E)$ with $l^\top a = L$, since we may otherwise decrease L to obtain an equivalent representation with smaller L. From $z_D^f(E) > 2$ we conclude $\mathbf{1} - a \notin P^p(E)$ so that $l^\top(\mathbf{1} - a) > L$. Thus $L < \frac{1}{2} l^\top \mathbf{1}$. Using $b^\top l \leq L$ for all $b \in P^p(E)$ and Eq. (3) we have

$$\frac{l^\top \mathbf{1}}{2} \sum_{b \in P^p(E)} \gamma_b > \sum_{b \in P^p(E)} \gamma_b \cdot L \geq \sum_{b \in P^p(E)} \gamma_b \cdot b^\top l = \mathbf{1}^\top l,$$

so that $z_C^p(E) = \sum_{a \in P^p(E)} \gamma_a > 2$. $\square$

Corollary 3. *For $z_D^f(E) = 3$ we have $\Delta_p(E) < 1$.*

For $z_D^f(E) = 4$ we can also conclude $z_C^p(E) > 2$ and $\Delta_p(E) < 2$ from $z_D^f(E) \leq \frac{4}{3} \cdot \lceil z_C^p(E) \rceil$, see [2] for the latter relative bound.

We remark that the instances with $z_D^p(E) \in \{n-1, n\}$ can be easily characterized. Since their number is in $O(n)$, we abstain from stating the details and provide exemplary enumeration results for $n = 8$ in Table 1.

6. Integer linear programming approaches

Assume that we are not interested in all equivalence classes of 1CSP but only in those with $\Delta_p(E) \geq \delta$ for some parameter $\delta \geq 0$. For the search for proper non-IRUP instances we may set $\delta = 1$ and for the search of the largest possible $\Delta_p(E)$ for a given demand n we may update δ during a search algorithm. In the following subsections we present two algorithmic approaches.

6.1. A tailored branch-and-bound algorithm

We can easily convert the enumeration algorithm from Section 4 into a branch-and-bound algorithm with some additional cuts. To this end we state:

Lemma 9. *Let $E = (n, L, l)$ be an 1CSP instance and U, V be two subsets of $\mathbb{B}^n$ such that $V \subseteq P^p(E) \subseteq U$. We have $z_C(U, E) \leq z_C(V, E)$ and $z_D(U, E) \leq z_D(V, E)$.*

Proof. Each feasible solution for pattern set V, i.e., each vector x with $A(V)x = \mathbf{1}$, can be extended to a feasible solution for pattern set U by setting $x_a = 0$ for all patterns a in $U \backslash V$. $\square$

Corollary 4. *If $V \subseteq P^p(E) \subseteq U \subseteq \mathbb{B}^n$ for an 1CSP instance $E = (n, L, l)$, then $\Delta_p(E) \leq z_D(V, E) - z_C(U, E)$.*

Our first modification of the enumeration algorithm is the extension of the lines 1 and 2 in the `RecursiveProcedure` with the check from Corollary 4.

Depending on the chosen value of δ we can also utilize some of the bounds from Section 5 to start the algorithm with a non-empty set $P_>$. For, e.g., $\delta \geq 1$ we know $z_D^f(E) > 3$ so that we can set $P_> = \{a \in \mathbb{B}^n : \|a\|_1 = n-2\}$ and remove each pattern $a \in \mathbb{B}^n$ with $\|a\|_1 \geq n-2$ from P_u. Even more, every insertion of a pattern a into $P_\leq$ may force some patterns to be non-feasible. If we can assume $z_D^f(E) > 2$ then, in particular, we have that $1-a$ is non-feasible and can be put into $P_>$. Moreover, all patterns $b \succeq 1-a$ are non-feasible and can be removed from P_u. As remarked, $\delta \geq 1$ implies $z_D^f(E) > 3$, so that $1-c$ has to be non-feasible whenever there are feasible patterns a, b with $a+b=c$. Again, all patterns $a' \succeq 1-c$ are non-feasible too and can be removed from P_u.

Because of the huge number of potential equivalence classes, the strategy of choosing pattern a from P_u in line 6 of the `RecursiveProcedure` is really important. The best branching strategy we found is to choose a pattern $a \in P_u$ with the maximum positive multiplier x_a in an optimal solution x of $z_C(U,E)$ for the set of patterns $U = P_u \cup \left\{a \in \mathbb{B}^n : \exists b \in P_\leq : a \preceq b\right\}$. Sometimes the optimal solution has no intersection with P_u. In this case we can choose the branching pattern at random. Indeed this happens in less than 0.01% of all cases. This strategy reduces the search space of about 1000 times in comparison to a random choice.

The B&B algorithm presented above was implemented in C++, where we used a self implemented LP-solver with exact arithmetic. Making use of Streaming SIMD Extensions (SSE4[2]) and special shortcuts for our LP instances, our implementation of an LP-solver is about 30 times faster than the COIN-OR LP-solver.[3] As hardware we have used an Intel Core i7 with 4 GB RAM.

6.2. A direct integer linear programming formulation

Instead of implementing a tailored B&B algorithm one can also formulate the problem of the maximization of $\Delta_p(E)$ for a given demand n as an integer programming problem and use off-the-shelf ILP solvers. To this end we describe the set $P^p(E)$ of proper patterns by binary variables $y_a \in \mathbb{B}$ for all $a \in \mathbb{B}^n$ and identify $P^p(E) = \{a \in \mathbb{B}^n : y_a = 1\}$. Partial information about $P^p(E)$ and $\mathbb{B}^n \backslash P^p(E)$ can be encoded by setting the variables of the respective patterns to either 1 or 0, respectively. Using the definition of an 1CSP instance only, we require $y_{\mathbf{0}} = 1$ and $y_{e_i} = 1$ for all $1 \leq i \leq n$.

To ensure the existence of feasible parameter values of L and l_i, $1 \leq i \leq n$, we have to further restrict the y_a-variables. Given an upper bound M on L, the inequalities of Lemma 1 can be formulated using so-called Big-M constraints. Fortunately all this is already known in the context of weighted simple games, see e.g. [12,14], i.e., each weighted game admits a representation where L can be upper bounded. So, without any further justification we state that the 1CSP instances with demand n are in one-to-one correspondence to the feasible 0/1 solutions y of:

$$
\begin{aligned}
& y_{\mathbf{0}} = 1 \\
& y_{e_i} = 1 \quad \forall\, 1 \leq i \leq n \\
& y_a - y_b \geq 0 \quad \forall\, a, b \in \mathbb{B}^n : a \preceq b \\
& \sum_{i: a_i = 1} l_i \leq L + (1 - y_a) \cdot M \quad \forall a \in B^n \\
& \sum_{i: a_i = 1} l_i \geq L + 1 - y_a \cdot M \quad \forall a \in B^n \\
& l_i \leq l_{i+1} \quad \forall\, 1 \leq i < n \\
& l_n \leq L \\
& y_a \in \mathbb{B} \quad \forall a \in \mathbb{B}^n \\
& L, l_i \in \mathbb{Z}_{\geq 1} \quad \forall\, 1 \leq i \leq n,
\end{aligned}
$$

where M can be chosen as $4n\left(\frac{n+1}{4}\right)^{(n+1)/2}$.

In principle we would like to maximize the target function $z_D^f(E) - z_C^p(E)$. Unfortunately both terms are the optimal values of optimization problems itself. Since the latter term arises from an LP we can model optimality by using the duality theorem, see [7] for an application of this technique in the context of simple games. Here it is even simpler since we can even take any feasible solution of the LP due to the maximization. So we replace $z_C^p(E)$ by $\sum_{a \in \mathbb{B}^n} x_a$ and add the constraints

$$
\begin{aligned}
& \sum_{a \in \mathbb{B}^n :\, a_i = 1} x_a = 1 \quad \forall\, 1 \leq i \leq n \\
& x_a \leq y_a \quad \forall\, a \in \mathbb{B}^n \\
& x_a \in \mathbb{R}_{\geq 0} \quad \forall\, a \in \mathbb{B}^n.
\end{aligned}
$$

---

2 See http://www.intel.com/support/processors/sb/CS-030123.htm.

3 Cf. [11], where the author also uses a self implemented LP solver to enumerate the weighted simple games with $n = 9$ voters.

For $z_D^f(E)$ this approach does not work, since there is no duality theorem for ILPs and $z_D^f(E)$ has a different sign as $z_C^p(E)$ in the target function. So we choose a different approach. We loop over all $k \in \{1, \ldots, n\}$, i.e., the theoretically possible values of $z_D^f(E)$, and we replace $z_D^f(E)$ by k in the target function. This gives us n different ILPs, i.e., one for each value of k. In the remaining part of the description we assume that k is fixed, i.e., we consider just one specific ILP. By introducing further inequalities we can ensure that $z_D^f(E) \geq k$ holds for all feasible solutions.

To this end let y_a^i be additional binary variables, which equals 1 if pattern $a \in \mathbb{B}^n$ can be written as the sum of at most $1 \leq i < k$ proper patterns a^j, where $j = 1, \ldots, i$. For $i = 1$ we have $y_a^1 = y_a$ for all $a \in \mathbb{B}^n$. Next we require $y_a^i \geq y_a^{i-1}$ for all $a \in \mathbb{B}^n$, $2 \leq i < k$ and

$$y_a^i \geq y_u^{i-1} + y_v^1 - 1 \quad \forall\, a, u, v \in \mathbb{B}^n : u + v = a \text{ and } \forall\, 2 \leq i < k.$$

As a justification let us consider a pattern $a \in \mathbb{B}^n$ that can be written as the sum of at most i proper patterns. If there exists such a representation with at most $i - 1$ summands, i.e. $y_a^{i-1} = 1$, then $y_a^i \geq y_a^{i-1}$ implies $y_a^i = 1$. Otherwise there exist a proper pattern v and a pattern u, that can be written as the sum of at most $i - 1$ proper patterns, with $a = u + v$. Thus, $y_u^{i-1} = 1$, $y_v^1 = 1$, and so also $y_a^i = 1$. We remark that $y_a^i = 1$ is also possible, if pattern a cannot be written as the sum of at most i proper patterns.

With these extra variables at hand, requiring $y_{(1,\ldots,1)}^{k-1} = 0$ guarantees $z_D^f(E) \geq k$. Note that a sum $\sum_{j=1}^{i} a^j \geq a$ of proper patterns a^j implies the existence of proper patterns $\tilde{a}^j$ with $\sum_{j=1}^{i} \tilde{a}^j = a$.

Many of these inequalities are redundant, as quickly found out by a customary ILP solver. We remark that it is not necessary to consider variables y_a^i for all $i \in \{1, \ldots, k-1\}$. By using inequalities of the form $y_a^i \geq y_u^{i_1} + y_v^{i_2} - 1$, where $i_1 + i_2 = i$, $i_1, i_2 < i$, $\Theta(\log k)$ values for i are sufficient in general. Nevertheless the ILP model becomes quite huge, so that we solved it with the Gurobi ILP solver on an Intel Xeon with 384 GB RAM. Of course one may try to deploy more sophisticated ILP techniques like column generation or cut separation, which goes beyond the scope of this paper.

7. Computational results

For $n \leq 9$ we have used the enumeration algorithm from Section 4 to generate all 1CSP instances with demand n. In Table 2 we have stated the number $\left|\mathbb{P}_n^p\right|$, the maximum value $\Delta_p(E)$, the number and the corresponding list of instances (representatives of equivalence classes) attaining this maximum value, whenever computationally possible. For each mentioned instance we have used the smallest possible integer valued parameters l_i and L.

The stated results for $n = 10, 11$ are obtained with the B&B-algorithm of Section 6.1 setting δ to 1 or $1 + \varepsilon$, respectively.[4] The computation time for $n = 11$ was 17 h.

For the cases $12 \leq n \leq 14$ we restricted the search to classes with large values of $z_D^p(E)$ due to the exponential growth of $\left|\mathbb{P}_n^p\right|$. For $n = 12$ we checked all equivalence classes with $z_D^p(E) \geq 5$, for $m = 13$ only those with $z_D^p(E) \geq 6$, and for $m = 14$ only those with $z_D^p(E) \geq 7$.

Using the ILP formulation of Section 6.2 (and suitable bounds from Section 5) we have verified the maximum Δ_p-value for $n \leq 11$, while consuming a considerably larger amount of computation time.

By slightly modifying the constraints of Lemma 2, according to the remarks in Section 3.1, we have also computed the number of weighted simple games with up to $n = 9$ voters. This uncovers read–write disk failures within the computation done in [11], so that the number of weighted simple games for $n = 9$ voters was corrected from 989 913 344 to 993 061 482.

8. Conclusion

We have presented an enumeration algorithm for all equivalence classes of 1CSP instances. For a demand of at most 9 the corresponding numbers are determined. As a side result we could correct the number of weighted simple games for 9 voters (incorrectly) stated in [11]. To the best of our knowledge, the relation between 1CSP instances and weighted simple games is indicated for the first time. By enhancing the enumeration approach to a B&B algorithm we were able to computationally prove that all 1CSP instances with demand of at most 9 pieces are proper IRUP instances, while we found classes of non-IRUP instances with demand $n = 10$ and $\Delta_p = 1$. This resolves an open question from [3,4], where the authors ask for proper non-IRUP instances with $n < 13$. $\Delta_p > 1$ is possible for $n \geq 11$ only. Even more, we have exactly determined the maximum proper gap Δ_p for $n \leq 11$ and classified all instances attaining the maximum gap. For further investigations on the structure of 1CSP instances with large gap, we have made them available at http://www.math.tu-dresden.de/~capad/capad.html. By partially going through the search space for $n \geq 12$, we improved the largest known proper gap from 1.003 to 1.0625.

With respect to the exact value $z_D^p(E)$ we have proven that all 1CSP instances with $z_D^p(E) \leq 3$ are proper IRUP instances with a proper gap smaller than 1, while there are examples with $z_D^f(E) = 4$ having $\Delta_p(E) > 1$.

Focusing on the size of L, and so indirectly on the size of the l_i, we mention that the first known constructions of proper non-IRUP instances were rather huge. The example of [16] has $L = 3, 397, 386, 355$ and was decreased to just

[4] Each ε with $0 < \varepsilon \leq \frac{1}{125}$ would have worked.

Table 2
Results of computational experiments.

n	$\lvert\mathbb{P}_n^p\rvert$	$\max \Delta_p$	[a]	Instances from classes with maximum Δ_p
1	1	0	1	$L = 1, l = (1)$
2	2	0	2	$L = 1, l = (1, 1)$; $L = 2, l = (1, 1)$
3	5	1/2	1	$L = 2, l = (1, 1, 1)$
4	17	2/3	1	$L = 3, l = (1, 1, 1, 1)$
5	92	3/4	2	$L = 4, l = (1, 1, 1, 1, 1)$; $L = 4, l = (1, 1, 2, 2, 3)$
6	994	7/8	1	$L = 8, l = (1, 2, 2, 3, 4, 5)$
7	28 262	16/17	1	$L = 17, l = (2, 3, 4, 5, 6, 7, 8)$
8	2 700 791	38/39	1	$L = 39, l = (2, 5, 6, 8, 11, 14, 15, 18)$
9	990 331 318	103/104	2	$L = 104, l = (7, 12, 16, 19, 22, 27, 30, 36, 40)$ $L = 104, l = (11, 15, 18, 20, 24, 27, 28, 32, 34)$
10		1	365	$L = 81, l = (4, 6, 6, 9, 16, 29, 32, 37, 40, 62)$ $L = 89, l = (4, 6, 7, 10, 18, 32, 35, 41, 44, 68)$ $L = 101, l = (5, 7, 8, 11, 20, 36, 40, 46, 50, 78)$ $L = 142, l = (7, 10, 11, 16, 28, 51, 56, 65, 70, 108)$ and 361 other instances
11		126/125	6	$L = 155, l = (9, 12, 12, 16, 16, 46, 46, 54, 69, 77, 102)$ $L = 193, l = (11, 15, 15, 20, 20, 57, 58, 67, 86, 96, 127)$ $L = 204, l = (12, 16, 16, 21, 21, 60, 61, 71, 91, 101, 134)$ $L = 207, l = (12, 16, 16, 21, 22, 61, 62, 72, 92, 103, 136)$ $L = 218, l = (13, 17, 17, 22, 23, 64, 65, 76, 97, 108, 143)$ $L = 221, l = (13, 17, 17, 23, 23, 65, 66, 77, 98, 110, 145)$
12		31/30[b]		$L = 18, l = (4, 4, 6, 6, 6, 7, 7, 9, 9, 9, 10, 12)$
13		53/50[b]		$L = 34, l = (8, 8, 10, 11, 11, 12, 12, 13, 13, 17, 17, 17, 18)$ $L = 48, l = (11, 11, 14, 15, 16, 17, 17, 18, 19, 24, 24, 24, 25)$
14		17/16[b]		$L = 42, l = (7, 7, 10, 10, 12, 15, 15, 21, 21, 21, 22, 22, 28, 31)$ $L = 50, l = (8, 9, 12, 12, 14, 18, 18, 25, 25, 25, 26, 26, 33, 37)$

[a] Number of classes with maximum Δ_p.
[b] Maximum found gap, without computational proof of optimality.

$L = 1, 111, 139$ in [2]. Recently the authors of [3,4] gave an example with $L = 100$. Our smallest found example has $L = 18$. It would be nice to know whether this is best possible.

We leave the famous (proper) MIRUP conjecture still widely open and encourage more research in that direction.

With respect to the enumeration of weighted simple games, the case of $n = 10$ voters might be in range of the presented exhaustive algorithm if further tuned. Some adaptation towards the inverse power index problem, see [12–14], is imaginable too.

Acknowledgments

The authors would like to thank two anonymous referees for very valuable remarks on an earlier draft. The research was supported by the Russian Foundation of Basic Research (RFBR), project 12-07-00631-a.

References

[1] S. Baum, L.E. Trotter Jr., Integer rounding for polymatroid and branching optimization problems, SIAM J. Algebraic Discrete Methods 2 (4) (1981) 416–425.
[2] L.M.A. Chan, D. Simchi-Levi, J. Bramel, Worst-case analyses, linear programming and the bin-packing problem, Math. Program. 83 (1–3) (1998) 213–227.
[3] J.C Díaz Díaz, Exact and heuristic solutions for combinatorial optimization problems (PhD thesis), Università Degli Studi di Modena e Reggio Emilia, 2012.
[4] A. Caprara, M. Dell'Amico, J.C. Díaz Díaz, M. Iori, R. Rizzi, Friendly bin packing instances without integer round-up property, Math. Program. Ser. A B (2014).
[5] F. Eisenbrand, D. Pálvölgyi, T. Rothvoß, Bin packing via discrepancy of permutations, ACM Trans. Algorithms (TALG) 9 (3) (2013) 24.
[6] M. Fieldhouse, The duality gap in trim problems, SICUP Bull. 5 (4) (1990).
[7] J. Freixas, S. Kurz, On α-roughly weighted games, Internat. J. Game Theory 43 (3) (2014) 659–692.
[8] P.C. Gilmore, R.E. Gomory, A linear programming approach to the cutting-stock problem, Oper. Res. 9 (6) (1961) 849–859.
[9] J.R. Isbell, A class of majority games, Q. J. Math. 7 (1) (1956) 183–187.
[10] V.M. Kartak, Sufficient conditions for the integer round-up property to be violated for the linear cutting stock problem, Autom. Remote Control 65 (3) (2004) 407–412.

[11] S. Kurz, On minimum sum representations for weighted voting games, Ann. Oper. Res. 196 (1) (2012) 361–369.
[12] S. Kurz, On the inverse power index problem, Optimization 61 (8) (2012) 989–1011.
[13] S. Kurz, The Inverse Problem for Power Distributions in Committees, Technical report, University of Bayreuth, 2013, 46 pages. Available at http://arxiv.org/abs/1402.0988.
[14] S. Kurz, S. Napel, Heuristic and exact solutions to the inverse power index problem for small voting bodies, Ann. Oper. Res. 215 (1) (2014) 137–163.
[15] O. Marcotte, The cutting stock problem and integer rounding, Math. Program. 33 (1) (1985) 82–92.
[16] O. Marcotte, An instance of the cutting stock problem for which the rounding property does not hold, Oper. Res. Lett. 4 (5) (1986) 239–243.
[17] C. Nitsche, G. Scheithauer, J. Terno, Tighter relaxations for the cutting stock problem, European J. Oper. Res. 112 (3) (1999) 654–663.
[18] J. Rietz, S. Dempe, Large gaps in one-dimensional cutting stock problems, Discrete Appl. Math. 156 (10) (2008) 1929–1935.
[19] J. Rietz, G. Scheithauer, J. Terno, Families of non-irup instances of the one-dimensional cutting stock problem, Discrete Appl. Math. 121 (1) (2002) 229–245.
[20] J. Rietz, G. Scheithauer, J. Terno, Tighter bounds for the gap and non-irup constructions in the one-dimensional cutting stock problem, Optimization 51 (6) (2002) 927–963.
[21] G. Scheithauer, J. Terno, About the gap between the optimal values of the integer and continuous relaxation one-dimensional cutting stock problem, in: Operations Research Proceedings 1991, Springer, 1992, pp. 439–444.
[22] G. Scheithauer, J. Terno, Theoretical investigations on the modified integer round-up property for the one-dimensional cutting stock problem, Oper. Res. Lett. 20 (2) (1997) 93–100.
[23] G. Scheithauer, J. Terno, The modified integer round-up property of the one-dimensional cutting stock problem, European J. Oper. Res. 84 (3) (1995) 562–571.
[24] G. Scheithauer, J. Terno, A branch-and-bound algorithm for solving one-dimensional cutting stock problems exactly, Appl. Math. 23 (2) (1995) 151–167.
[25] A.D. Taylor, W.S. Zwicker, Simple Games: Desirability Relations, Trading, Pseudoweightings, Princeton University Press, 1999.