ARTICLE IN PRESS
JID: EOR [m5G;July 9, 2018;8:56]

European Journal of Operational Research 000 (2018) 1–15

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Improved state space relaxation for constrained two-dimensional guillotine cutting problems

André Soares Velasco[a], Eduardo Uchoa[b,*]

[a] Instituto Federal Fluminense, IFF Av. Souza Mota, 350. Parque Fundão, Campos dos Goytacazes RJ. CEP: 28060-010, Brazil
[b] Departamento de Engenharia de Produção, Universidade Federal Fluminense, UFF Rua Passo da Pátria 156, Bloco D. São Domingos, Niterói RJ. CEP:24210-240, Brazil

ARTICLE INFO

Article history:
Received 14 July 2017
Accepted 7 June 2018
Available online xxx

Keywords:
Cutting
Dynamic programming
Integer programming

ABSTRACT

Christofides and Hadjiconstantinou (1995) introduced a dynamic programming state space relaxation for obtaining upper bounds for the Constrained Two-dimensional Guillotine Cutting Problem. The quality of those bounds depend on the chosen item weights, they are adjusted using a subgradient-like algorithm. This paper proposes Algorithm X, a new weight adjusting algorithm based on integer programming that provably obtains the optimal weights. In order to obtain even better upper bounds, that algorithm is generalized into Algorithm X2 for obtaining optimal two-dimensional item weights. We also present a full hybrid method, called Algorithm X2D, that computes those strong upper bounds but also provides feasible solutions obtained by: (1) exploring the suboptimal solutions hidden in the dynamic programming matrices; (2) performing a number of iterations of a GRASP based primal heuristic; and (3) executing X2H, an adaptation of Algorithm X2 to transform it into a primal heuristic. Extensive experiments with instances from the literature and on newly proposed instances, for both variants with and without item rotation, show that X2D can consistently deliver high-quality solutions and sharp upper bounds. In many cases the provided solutions are certified to be optimal.

1. Introduction

The Two-dimensional Guillotine Cutting Problem (TGCP) consists in determining the most valuable way of cutting a rectangular object with length L and width W, using only orthogonal guillotine cuts, in order to produce smaller rectangular pieces, that are copies of m distinct items with predefined dimensions and value. For $1 \leq i \leq m$, l_i denotes the length of i, w_i its width and v_i its value. A guillotine cut must cross the object, or a rectangular shape obtained from previous cuts, from one side to the other. Some authors also refer to that problem as the Guillotine Two-dimensional Knapsack Problem. The Constrained TGCP (CTGCP) is the generalization where each item i also has a given demand D_i, the maximum number of copies of an item in the cutting pattern. Orthogonal Rotations of items in the cutting patterns may be permitted or not. This work addresses both variants: CTGCP with rotation and without rotation. Cuts are often performed in stages, where each stage consists of a set of parallel guillotine cuts on the rectangular shapes obtained in the previous stages. We assume that there are no restrictions on the number of stages. Sometimes the value of an item is defined by its area, the so called unweighted case. We also consider the case known as weighted, where item values are arbitrary.

The CTGCP is a classical problem with many industrial applications. For example, the objects to be cut may be glass or wood panels, metal sheets, marble or granite slabs, etc. While TGCP can be solved in pseudo-polynomial time by Dynamic Programming (DP) (Gilmore & Gomory, 1965), CTGCP is known to be strongly NP-hard (Hifi, 2004) and can be much harder in practice. The proposed exact methods for CTGCP are listed next. The first ones were combinatorial branch-and-bound algorithms, like those in Christofides and Whitlock (1977), Christofides and Hadjiconstantinou (1995), Cung, Hifi, and Le Cun (2000). More recently, Chen (2008) proposed an enumerative algorithm. The idea is to explore the solution space recursively, an exponential procedure that should be accelerated by the use of effective bounds for pruning unpromising branches. Dolatabadi, Lodi, and Monaci (2012) also presented enumerative algorithms, A1 and A2. Computational experiments show that Algorithm A1 has a better performance and can solve many classical instances to optimality. That algorithm receives external lower and upper bounds, $ExtZ_{LB}$ and $ExtZ_{UB}$. It performs its recursive search of the solution space by only looking for solutions

* Corresponding author.
E-mail addresses: asvelasco@iff.edu.br (A.S. Velasco), uchoa@producao.uff.br (E. Uchoa).

https://doi.org/10.1016/j.ejor.2018.06.016
0377-2217/© 2018 Elsevier B.V. All rights reserved.

with value equal or superior to z_0. Initially, z_0 is set to $ExtZ_{UB}$. If no solution is found, then z_0 is decreased and another search is performed. Furini, Malaguti, and Thomopulos (2016) created the first practical MIP models for the CTGCP without stage limit, which is capable of solving instances of reasonable size. Recently, Fleszar (2016) proposed an algorithm for the related problem of deciding whether all items of a given set can be obtained from a rectangular object by guillotine cuts.

Heuristics for CTGCP were also proposed. Alvarez-Valdés, Parajón, and Tamarit (2002) presented tabu search and GRASP metaheuristics, both complemented by path relinking procedures. Hifi (2004) gives a hybrid algorithm that uses dynamic programming and a hill-climbing procedure. A class of heuristics of particular interest is the primal-dual heuristic, where a dual method (able to find upper bounds on the optimal solution value) is adapted for also finding primal feasible solutions (that yield lower bounds on the optimal solution value). Morabito and Pureza (2010) proposed a primal-dual heuristic and showed that it can find some solutions that are difficult to be found using pure primal heuristics, like metaheuristics. By their own nature, primal-dual heuristics provide solutions together with an upper bound. Sometimes the upper bound matches the solution value, certifying that it is indeed optimal.

The best known upper bounds for the CTGCP that can be obtained in pseudo-polynomial time are usually those by the DP State Space Relaxation (DPSSR), introduced in Christofides and Hadjiconstantinou (1995). The DPSSR is based in the following idea:

- The DP for the TGCP cannot be turned into an efficient exact algorithm for CTGCP, since that would require adding up to m dimensions to its recursion, leading to an exponential explosion in the number of states. Instead, they propose a DP recursion with a single additional dimension that can be viewed as a relaxation of the exact recursion: a non-negative integer weight q_i is associated to each item i and it is imposed that the sum of the weights of the items in a solution should not exceed $Q = \sum_{i=1}^{m} (D_i q_i)$.

The upper bound actually provided by DPSSR depends on the chosen weights. Christofides and Hadjiconstantinou (1995) proposed an iterative procedure where all weights start with value zero and are adjusted by a subgradient-like formula. Morabito and Pureza (2010) used DPSSR as the basis of their primal-dual heuristic DP_AOG and proposed an improved formula for weight adjusting. *The main contribution of this paper is Algorithm X, an alternative algorithm for weight adjusting in DPSSR. Algorithm X is based on an integer programming model and is proved to be optimal, in the sense that it finds the weights that yield the best possible upper bound obtainable by DPSSR.* Other important contributions are:

- A generalized variant of the DPSSR that uses two-dimensional item weights for obtaining even stronger upper bounds. Algorithm X2, also based on integer programming, for the optimal adjustment of those weights is proposed.
- A primal-dual heuristic, called X2D. It executes Algorithms X and X2, but also uses a number of additional methods for obtaining good feasible solutions:
  - The suboptimal solutions hidden in the dynamic programming matrices are explored. While the optimal DPSSR solution can only be feasible if it is also the optimal CTGCP solution, suboptimal DPSSR solutions can be good feasible CTGCP solutions. Moreover, "near-feasible" solutions obtained from those matrices can often be corrected into good feasible solutions by performing local substitutions.
  - On instances where the gaps between upper and lower bounds are still large ($>0.3\%$), a number of iterations of a GRASP-based pure primal heuristic (Velasco & Uchoa, 2014) are performed.
  - Finally, Algorithm X2H, an adaptation of Algorithm X2 to transform it into a primal heuristic, may be executed.

We report extensive computational experiments on 500 instances from the literature. For the variant without rotation, X2D is compared with the best heuristic (Morabito & Pureza, 2010) and the best exact algorithm (Dolatabadi et al., 2012) available in the literature. We also report results for the CTGCP with rotation. In that case, there are no recent algorithms in the literature for comparisons. Anyway, for both with or without rotation variants, we show that X2D can consistently deliver high-quality solutions and sharp upper bounds in reasonable times. The provided solutions are often certified to be optimal. We also present experiments on 80 newly created hard instances.

The article is organized as follows. Section 2 describes the existing DPSSR. Sections 3 and 4 present Algorithms X and X2, respectively. Section 5 describes the primal components used in primal-dual heuristic X2D. Section 6 presents computational results. The last section presents final remarks. Additional computational results and comparisons for other instances from the literature are available as Supplementary Material. For simplicity, all proposed algorithms will assume the variant without rotation. However, in Section 6, we indicate how they can be easily adapted for the variant with rotation.

2. Dynamic programming state space relaxation for the CTGCP

An optimal solution for the TGCP can be assembled from the optimal solutions of the two subproblems defined by each possible horizontal or vertical guillotine cut. This fact allows its solution in pseudo-polynomial time by Dynamic Programming (DP), the complexity depends on the values of L and W. The original recursion proposed by Gilmore and Gomory (1965) limits the maximum number of cutting stages. Of course, it can be used to obtain the optimal solution without restriction on the number of stages by setting a sufficiently large stage limit. Beasley (1985) gives a simpler recursion for the case without any stage limit. That recursion, together with the concept of Discretization Points from (Herz, 1972), was used by Cintra, Miyazawa, Wakabayashi, and Xavier (2008) on developing an exact DP algorithm for TGCP that is very effective when the values of L and W are not too large. Instances with values of L and W around 100 are solved in milliseconds; instances with L and W around 1000 can be solved in a few seconds in a modern computer.

On the other hand, solving the CTGCP by DP is a much more demanding task. This is related to the need of controlling how many copies of each item appear in the solutions of each subproblem. Let $\mathbf{D} = [D_1, \ldots, D_m]$ and $\mathbf{C} = [C_1, \ldots, C_m]$ be integer vectors indicating the original demand and the maximum number of copies of each item allowed in the solution of a subproblem, respectively. Define

$$v(l, w, \mathbf{C}) = \max(\{v_i | 1 \le i \le m : l_i \le l, w_i \le w, C_i \ge 1\} \cup \{0\}) \quad (1)$$

as the maximum value that can be obtained by cutting, without rotation, a single piece of an item i with positive C_i from a rectangle with dimensions (l, w). The value of the best solution for a rectangle (l, w), respecting the limits indicated by $\mathbf{C}$, is obtained by:

$$V(l, w, \mathbf{C}) = \max \begin{cases} \{v(l, w, \mathbf{C})\} \cup \\ \{V(l', w, \mathbf{C}') + V(l - l', w, \mathbf{C} - \mathbf{C}') \mid l' \in P_1, l' \le l/2, \mathbf{0} \le \mathbf{C}' \le \mathbf{C}\} \cup \\ \{V(l, w', \mathbf{C}') + V(l, w - w', \mathbf{C} - \mathbf{C}') \mid w' \in P_2, w' \le w/2, \mathbf{0} \le \mathbf{C}' \le \mathbf{C}\} \end{cases} \quad (2)$$

This means that $V(l, w, \mathbf{C})$ is either $v(l, w, \mathbf{C})$ or is the sum of the best solutions of the two subproblems defined by some vertical or horizontal guillotine cut and by some way of splitting C. The value