Int. J. Production Economics 144 (2013) 432–439

Contents lists available at SciVerse ScienceDirect

Int. J. Production Economics

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

Sequential grouping heuristic for the two-dimensional cutting stock problem with pattern reduction

Yaodong Cui *, Liu Yang, Zhigang Zhao, Tianbing Tang, Mengxiao Yin

School of Computer, Electronics and Information, Guangxi University, 100 Daxue Road, Nanning 530004, China

ARTICLE INFO

Article history:
Received 10 January 2012
Accepted 8 March 2013
Available online 21 March 2013

Keywords:
Two-dimensional cutting stock
Sequential heuristic procedure
Sequential value correction method
Pattern reduction
Sequential grouping heuristic

ABSTRACT

A sequential grouping heuristic (SGH) that supports parallel computing is presented for solving the two-dimensional cutting stock problem with pattern reduction, where a set of rectangular items with given demand are cut from rectangular stock plates of the same size, considering both input-minimization (main objective) and pattern reduction (auxiliary objective). It is based on the sequential heuristic procedure that generates each next pattern to fulfill some portion of the remaining items and repeats until all items are fulfilled. The SGH uses a grouping technique to select the items that can be used to generate the next pattern, and adjusts the item values according to the sequential value correction method after the next pattern is generated. Each next pattern is generated using a dynamic programming recursion. The computational results indicate that the SGH is powerful in both input-minimization and pattern reduction, and the parallel computing is useful to reduce computation time.

1. Introduction

In the general two-dimensional cutting stock problem (2DCSP), a set of rectangular item types with given size and demand are cut from rectangular stock plates of the same size, such that the plate count is minimized (input-minimization). The formal term for the 2DCSP is the two-dimensional rectangular single stock size cutting stock problem (Wäscher et al., 2007). The solution of the 2DCSP is a cutting plan that contains a set of cutting patterns. Each pattern is specified with a frequency and the numbers of each item type included. The included items should not overlap and the pattern should be feasible for the cutting process.

Restrictions and objectives with practical backgrounds are often addressed in solving cutting stock problems (Arbib et al., 2012; Erjavec et al., 2012). Although the main objective in solving the 2DCSP is input-minimization, it is necessary to reduce the pattern count (pattern reduction) of the cutting plan for some applications. Pattern reduction is useful at least for the following aspects:

1. Each pattern is often related with a setup cost in the cutting process. Pattern reduction is useful to reduce the total setup cost.
2. For some industries (such as the wood and furniture industries), sheets of the same pattern may be piled to cut. Pattern reduction can lead to lower cutting cost because the number of piles can be reduced.

This paper considers the following two-dimensional guillotine cutting stock problem with pattern reduction (2DCSPPR): m rectangular item types with given demand are cut from rectangular stock plates of size $L \otimes W$ (length⊗width) using guillotine cuts, where the main objective is input-minimization, the auxiliary objective is pattern reduction, and the two objectives are considered in priority mode. The length, width and demand of type-i items are l_i, w_i and d_i, respectively. The items are orientated, that is, an item can only be placed with the length in horizontal orientation. Let N be the pattern count of the cutting plan, P_j be the jth pattern that contains p_{ij} type-i items, x_j be the frequency of P_j, and $\mathbf{\Omega}$ be the set of non-negative integers. The solution to the 2DCSP can be determined from solving the following integer programming formulation:

$$Z=\min\sum_{j=1}^{N} x_j;\quad \sum_{j=1}^{N} p_{ij}x_j = d_i;\quad i=1,\dots,m;\quad x_j\in\mathbf{\Omega};\quad j=1,\dots,N$$

Although exact methods can be used to solve the general 2DCSP, it may be difficult for them to consider multiple objectives and constraints, especially when the instance scale is large. A sequential grouping heuristic (SGH) is presented in this paper. It is based on the sequential heuristic procedure (SHP) that generates each next pattern to fulfill some portion of the remaining items and repeats until all items are fulfilled. The SGH adjusts the item values according to the sequential value correction method (Belov and Scheithauer, 2007), and uses a grouping technique to select the items that can be used in generating the

* Corresponding author. Tel.: +86 771 3232214; fax: +86 10 950507/704878.
E-mail address: ydcui@263.net (Y. Cui).

0925-5273/$ - see front matter © 2013 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.ijpe.2013.03.011

next pattern. Simple block patterns (Cui, 2007a) are used and a heuristic is presented for generating them. The SGH is extended to support parallel computing. The computational results indicate that the SGH is powerful in both input-minimization and pattern reduction.

The literature is reviewed in Section 2. Simple block patterns are introduced and a heuristic for generating them is presented in Section 3. The SGH is described in Section 4. The computational results are reported in Section 5, followed by the conclusions in Section 6.

Notations and abbreviations used across sections/sub-sections are listed in Appendix for quick reference. They are also introduced where they appear the first time in this paper.

2. Literature review

There are mainly three deterministic approaches for the 2DCSP. The first approach includes algorithms based on linear programming (LP), where the decision variables are the frequencies of the patterns. A computational study of LP-based heuristic algorithms is available in Alvarez-Valdes et al. (2002). It follows the column generation scheme in which at each cycle of the iteration, a new cutting pattern is generated as the solution of a sub-problem on one stock plate. In their work, three heuristic procedures of increasing complexity are used to solve the sub-problem, producing solutions differing in quality and computation time. Different rounding procedures are also compared to obtain integer frequencies of the patterns from the fractional solutions of the LP approach.

The work in Cintra et al. (2008) also uses several procedures to solve the sub-problem. The fractional frequencies of the patterns are round down to obtain integer solutions and two strategies are provided to tackle the residual instances. Vanderbeck (2001) presented a LP-based approach, where homogenous three-staged patterns (Cui, 2008) are used. The approach involves a nested decomposition of the problem and the recursive use of the column-generation technique.

The second approach is based on integer programming (IP). Belov and Scheithauer (2006) presented a branch-and-cut-and-price algorithm for the 2DCSP with two-staged patterns. The solution approach strengthens the LP relaxation at each branch-and-price node with mixed-integer cuts. The effectiveness of the approach is demonstrated through tests.

An integer programming model for the 2DCSP is available in Silva et al. (2010), where two- and three-staged patterns are used. The construction and optimization of the proposed model were implemented in C++ using the C callable library of the ILOG Cplex 11.0 solver.

An integer linear programming arc-flow model for the 2DCSP with two-staged patterns is available in Macedo et al. (2010), formulated as a minimum flow problem. It is an extension of the arc-flow model proposed by Valério de Carvalho (1999) for the one-dimensional cutting stock problem (1DCSP). The model was also solved with ILOG Cplex 11.0 solver.

The third approach is based on the SHP that generates the patterns in the cutting plan sequentially. It is welcomed because of its ability to control and consider factors (objectives and constraints) other than input-minimization. Although this type of algorithms has been widely used to solve the 1DCSP (Belov and Scheithauer, 2007; Cui and Liu, 2011), they have not been used frequently to solve the 2DCSP. The authors have not found one that solves the 2DCSPPR. Yanasse et al. (1991) presented a SHP for a cutting stock problem in the wood industry, where there are several different board sizes from which panels can be cut. Suliman (2006) presented a SHP for the case where rolls are used to produce the items, and the patterns belong to a simple type: one-stage homogenous strip cutting patterns (Cui and Yang, 2010). No computational results are reported except the solution to an illustrative example. An iterative SHP to a real-life cutting stock problem arising in a make-to-order plastic company is available in Song et al. (2006). The plates have different lengths and the same width. Customers may want items longer than the available plate length. Instead, several smaller ones are offered so that the total length of the items satisfies the demand.

It is difficult to find journal papers that deal with the 2DCSPPR. Imahori et al. (2005) presented a local search algorithm for the 2DCSP*n* (a special case of the 2DCSPPR) in a book chapter, where the plate count of the cutting plan should be minimized, observing the constraint that the pattern count must not exceed n.

To solve the 2DCSPPR, each pattern in the cutting plan should be generated from solving the bounded SLOPP (Single Large Object Placement Problem), in which a set of rectangular items are to be arranged on the stock plate without overlap, such that the pattern value (the total value of the included items) is maximized, where the frequency of each item type should not exceed the upper bound. Most algorithms assume that guillotine cuts are necessary to divide the plate into items. A guillotine pattern can be either general or restricted. The former considers only the restriction of the guillotine cuts. The latter observes both the guillotine cuts and other restrictions. The restricted patterns may be classified into two types. Patterns of the first type are established to simplify the cutting process. Two-staged, three-staged, T-shape and n-group patterns are among them. Patterns of the second type are proposed to simplify the design of the pattern-generation procedure and to reduce the solution space to make the computation time affordable. The simple block patterns used in this paper belongs to this type.

Exact algorithms for general guillotine patterns are available in Wang (1983), Viswanathan and Bagehi (1993), Daza et al. (1995), Hifi (1997), Cung et al. (2000), and Amaral and Wright (2001). They are often not practical for solving medium and large instances because of the long computation time.

Exact and heuristic algorithms for the first type of restricted patterns have been reported intensively. Hifi and Roucairol (2001), Hifi (2005), Hifi and M'Hallah (2006), Hifi et al. (2008), and Hifi et al. (2012) presented both exact and heuristic algorihtms for two-staged patterns. Cui (2012a) described a heuristic for T-shape patterns. He also published exact or heuristic algorithms for homogenous two-segment patterns (Cui, 2007b) and homogenous three-staged patterns (Cui, 2008), where a homogenous strip contains only items of the same size. Yanasse et al. described algorithms for one-group patterns (Yanasse and Morabito, 2006), two- and three-group patterns (Yanasse and Morabito, 2008), and checkerboard patterns (Yanasse and Katsurayama, 2008).

Other heuristic algorithms for solving the two-dimensional bounded SLOPP generate the second type of restricted patterns, if the geometric feature of the patterns is specified to reduce the solution space and to facilitate the design of the algorithms. The pattern-generation procedure of this paper is among them. The exact algorithm for the two-dimensional unbounded SLOPP in Cui (2007a) generates simple block patterns, where the computational results of 41 benchmark instances indicate that simple block patterns may lead to material utilization better than that of three-staged patterns and their variants. This is a reason to use simple block patterns in this paper. Another reason is that the pattern-generation procedure is simple and easy to code. This is helpful to practical applications.

Algorithms for solving the 1DCSP with pattern reduction (1DCSPPR) will be briefly reviewed in this paper. More detailed reviews are available in Cerqueira and Yanasse (2009), and Cui and Liu (2011). Vanderbeck (2000) formulated the problem as a