European Journal of Operational Research 000 (2014) 1–7

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Pattern-set generation algorithm for the one-dimensional cutting stock problem with setup cost

Yaodong Cui*, Cheng Zhong, Yi Yao

College of Computer and Electronic Information, Guangxi University, Nanning 530004, China

A R T I C L E I N F O

Article history:
Received 9 April 2014
Accepted 9 December 2014
Available online xxx

Keywords:
Cutting
Cutting stock
One-dimensional cutting
Pattern reduction
Setup cost

A B S T R A C T

The primary objective in the one-dimensional cutting stock problem is to minimize material cost. In real applications it is often necessary to consider auxiliary objectives, one of which is to reduce the number of different cutting patterns (setups). This paper first presents an integer linear programming model to minimize the sum of material and setup costs over a given pattern set, and then describes a sequential grouping procedure to generate the patterns in the set. Two sets of benchmark instances are used in the computational test. The results indicate that the approach is efficient in improving the solution quality.

1. Introduction

In the classical one-dimensional cutting stock problem (CSP), m item types with lengths $(l_1, \ldots, l_m)$ and demands $(d_1, \ldots, d_m)$ are cut from stock bars of length L to minimize material cost. The solution is a cutting plan that contains a set of different cutting patterns, each of which has specified frequency. The primary objective in the CSP is to minimize material cost. Auxiliary objectives (costs) exist in real-life production (Cherri, Arenales, Yanasse, Poldi, & Vianna, 2014; Kallrath, Rebennack, Kallrath, & Kusche, 2014), one of which is pattern reduction. It is necessary to adjust the positions of the cutting tools in the cutting machine each time a new pattern is set up. Reducing the pattern count (the number of different cutting patterns) of the cutting plan is useful to decrease setup cost, especially for the cases where setup cost is high. A typical case is the cutting of a large steel slab, where the cutting tools have large size, changing their positions incurs much additional cost and setup time.

A heuristic is presented in this paper for the CSP with pattern reduction (CSPPR). It solves the CSPPR in two stages. In the first stage, it calls a sequential grouping procedure (SGP) to generate a set of patterns. In the second stage, it uses the CPLEX optimizer to solve an integer linear programming (ILP) model that minimizes the sum of material and setup costs over the given pattern set. The proposed algorithm is called the SGPIP to denote that the SGP is used in the first stage to generate the patterns and an integer programing model (IP) is solved in the second stage over the patterns. Computational test is performed on two sets of benchmark instances to compare the algorithms for the CSPPR. The results show that the SGPIP yields the best average solution quality.

The remaining contents are arranged as follows. The literature is reviewed in the next section. The ILP model that minimizes the sum of material and setup costs over a given pattern set is established in Section 3, together with the general frame of the SGPIP. The SGP for generating the patterns in the pattern set is presented in Section 4. Computational results are reported in Section 5 and conclusions are given in the last section.

2. Literature review

A simple literature review for the CSPPR is given in this section. A similar review can be found in Cui and Liu (2011).

An exact approach is presented in Vanderbeck (2000). It formulates the CSPPR as a quadratic integer programming problem. The objective is to minimize the pattern count, given the minimum number of bars required to meet the item demands. Sixteen practical instances were used in the experimental test, where the number of item types ranges from 5 to 32. Only 12 instances were solved to optimality because of the computation time limit (2 h per instance). This indicates that the approach is adequate for solving small instances.

Approximate algorithms that are based on the sequential heuristic procedure (SHP) are widely used to solve the CSPPR. The SHP in Haessler (1975) selects a pattern that satisfies the aspiration levels of trim loss and frequency. It is based on the observation that increasing the frequency of the current pattern is often useful for pattern reduction. The SHP in Vahrenkamp (1996) uses the same idea, where the current pattern is chosen from 200 patterns obtained from random

* Corresponding author. Tel.: +8607713232214.
E-mail address: ydcui@263.net (Y. Cui).

http://dx.doi.org/10.1016/j.ejor.2014.12.015
0377-2217/

bin packing. The algorithms in Cui, Zhao, Yang, and Yu (2008) and Cui and Liu (2011) generate the current pattern using a subset of the remaining items. This subset is determined to maximize the frequency of the current pattern, observing the constraint that both the number of included item types and the total length of the included items should not be smaller than the specified thresholds. Dikili, Sariöz, and Pek (2007) solved the CSPPR in two-stages. A simple heuristic generates a large set of cutting patterns in the first stage. In the second stage, a SHP selects some of the first-stage patterns to form the cutting plan.

The heuristic approach in Foerster and Wäscher (2000) solves the CSPPR in two stages. A cutting plan is generated at the first stage without considering pattern reduction. A pattern combination method is used for pattern reduction at the second stage. It iteratively combines two patterns to 1, 3 to 2 and 4 to 3, until the pattern count cannot be reduced further.

The heuristic approach presented in Umetani, Yagiura, and Ibaraki (2003) allows both surplus and shortage of the item types. It cannot solve the CSPPR of this paper because the CSPPR does not allow shortage of the items and takes surplus as waste. Their approach and the computational results will not be further commented for this reason.

A hybrid heuristic method is available in Yanasse and Limeira (2006). It solves the CSPPR in three stages. A SHP generates some patterns in the cutting plan at the first stage, subject to the constraint that each pattern must fulfill the demands of at least two item types. A residual problem is formed after all such patterns have been generated. It is also solved by the SHP without considering the previous constraint. The patterns of the first two stages form the complete cutting plan. The pattern combination method in Foerster and Wäscher (2000) is used at the third stage to reduce the pattern count further.

Belov and Scheithauer (2003) proposed an integer programming model for pattern reduction, and developed a branch-and-price algorithm to solve the model. The computational results show that the algorithm outperforms that of Foerster and Wäscher (2000) in solution quality. Later the authors extended the algorithm to deal with both pattern reduction and open stacks minimization (Belov & Scheithauer, 2007).

Mobasher and Ekici (2013) developed a mixed integer linear program model and proposed two local search algorithms and a column generation based heuristic algorithm.

Cui, Yang, Zhao, Tang, and Yin (2013) presented a sequential grouping heuristic for the two-dimensional cutting stock problem with pattern reduction, where *sequential* means that the patterns in a cutting plan are generated sequentially and *grouping* indicates that each next pattern is obtained from considering only the items in a selected subset of the remaining items. The idea will be used in this paper to design the SGP to generate the patterns.

This paper formulates the CSPPR as an ILP that minimizes the sum of material and setup costs over a given pattern set. The proposed algorithm SGPIP first calls the SGP to generate the patterns in the set, and then uses the CPLEX optimizer to solve the ILP over the set. The approach is heuristic because not all possible patterns are considered. Computational test on two sets of benchmark instances is performed to compare the algorithms for the CSPPR. The results show that the SGPIP performs the best in improving the solution quality.

Recently, a pattern-set generation algorithm (PSG) for the one-dimensional multiple stock sizes cutting stock problem is presented in Cui, Cui, and Zhao (2014). It also generates a set of patterns in the first stage and solves an ILP model over the generated patterns in the second stage. It differs from the SGPIP mainly in the following aspects: (1) The PSG uses a residual heuristic to generate the patterns in the first stage, whereas the SGPIP uses the SGP to obtain the patterns. The computational results later reported in Section 5.3 indicate that the residual heuristic used by the PSG may not be effective in pattern reduction. (2) The ILP model solved by the PSG does not consider pattern reduction.

3. ILP model and general frame of the algorithm SGPIP

The following notations are used:

Z_{ILP}	objective value (sum of material and setup costs)
Q	set of patterns, $Q = \{Q_1, \ldots, Q_N\}$
N	number of patterns in Q
u_{b}	cost per bar
u_{s}	cost per setup (cost of each new pattern)
x_j	frequency of pattern Q_j
ε_j	0/1 variable denoting whether Q_j is used ($\varepsilon_j = 1$) or not ($\varepsilon_j = 0$)
q_{ij}	number of type$-i$ items in pattern Q_j
M	upper bound of pattern frequency, $M = \sum_{i=1}^{m} d_i$
I	set of non-negative integers

The ILP model for the CSPPR is as follows:

$$Z_{\text{ILP}} = \min\left(u_{\text{b}} \sum_{j=1}^{N} x_j + u_{\text{s}} \sum_{j=1}^{N} \varepsilon_j\right) \tag{1-1}$$

$$\sum_{j=1}^{N} q_{ij} x_j \ge d_i, \quad i = 1, \ldots, m \tag{1-2}$$

$$x_j \le M\varepsilon_j, \quad j = 1, \ldots, N \tag{1-3}$$

$$x_j \in I, \quad \varepsilon_j \in \{0, 1\}, \quad j = 1, \ldots, N$$

Formula (1-1) means that the objective is to minimize the sum of material and setup costs. Constraint (1-2) indicates that the item demands must be meet. Constraint (1)–(3) guarantees that pattern Q_j can be used only when $\varepsilon_j = 1$. Although $M = \sum_{i=1}^{m} d_i$ is used in the computational test of this paper, it can be defined depending on pattern j ($x_j \le M_j\varepsilon_j$, $M_j = \max\{\lceil d_i/q_{ij}\rceil | i : q_{ij} > 0\}$) to get possibly stronger restrictions.

The SGPIP of this paper solves the ILP model in two stages. In the first stage, it calls the SGP (described in the next section) to generate the patterns in Q. Meanwhile a stage-one solution is obtained. In the second stage, it uses the CPLEX optimizer as optimization engine to solve the ILP model to obtain the stage-two solution. The stage-two solution may be not optimal (over set Q) because a limit is placed on the computation time. Hence the better one of the two solutions is selected.

4. Procedure SGP for generating the patterns

4.1. Steps of the SGP

Procedure SGP is called by the SGPIP in the first stage to generate the patterns in set Q and to obtain the stage-one solution. The following notations are used to describe it:

Z	cost of the current cutting plan
Z_{SGP}	cost of the best cutting plan obtained in performing the SGP. It is also the cost of the stage-one solution when the SGP is finished
G	ID of the current generation
$G_{\max}$	number of maximum generations
P	current pattern, $P = (p_1, \ldots, p_m)$
r_i	remaining demand of type-i items, $i = 1, \ldots, m$
b_i	maximum number of type-i items that can be used for generating the current pattern P, $i = 1, \ldots, m$.
c_i	value of a type-i item
α_G	grouping parameter for generation G, $G = 1, \ldots, G_{\max}$.
β_G	grouping parameter for generation G, $G = 1, \ldots, G_{\max}$.

Please cite this article as: Y. Cui et al., Pattern-set generation algorithm for the one-dimensional cutting stock problem with setup cost, European Journal of Operational Research (2015), http://dx.doi.org/10.1016/j.ejor.2014.12.015