



Discrete Optimization

The minimization of open stacks problem: A review of some properties and their use in pre-processing operations

Horacio Hideki Yanasse^{a,*}, Edson Luiz França Senne^b^a Instituto Nacional de Pesquisas Espaciais – INPE/LAC, Avenida dos Astronautas 1758, Jardim da Granja, 12227-010 São José dos Campos, São Paulo, Brazil^b Universidade Estadual Paulista Júlio de Mesquita Filho – UNESP, Faculdade de Engenharia de Guaratinguetá, Av. Ariberto Pereira da Cunha, 333, Portal das Colinas, 12516-410 Guaratinguetá, SP, Brazil

ARTICLE INFO

Article history:

Received 18 June 2008

Accepted 15 September 2009

Available online 19 September 2009

Keywords:

Combinatorial optimization

Minimization of open stack problem

Pre-processing

Size reduction

ABSTRACT

Pre-processing operations that reduce the size of a problem may be decisive for solving or not solving practical instances of a NP-hard problem. In this article we review some properties suggested in the literature for the minimization of open stacks problem that can be used in pre-processing operations to reduce the instances sizes. We also present a new pre-processing technique that may be very effective in reducing the size of an instance. We present computational tests with the suggested pre-processing operations applied on sets of MOSP instances of the literature and we show that the reductions obtained can be significant.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Cutting stock problems arise in many productive settings, like in the paper, the furniture, the steel, the wooden hardboard industries. The problem consists in cutting smaller pieces (denoted items), with fixed dimensions and quantities, from larger pieces (denoted objects), with some optimization criterion, for example, maximizing profits or minimizing trim loss. A solution of a cutting stock problem is composed of a set of cutting patterns and their corresponding frequencies, that is, the number of times the patterns have to be cut in order to fulfill the demands of the desired items.

When cutting the patterns, the items cut are piled up in stacks, one stack for each item type. A stack is open the first time an item type is cut and it remains open until the last piece of the corresponding item type is cut. When the last piece is cut, the stack is closed. It is desirable to maintain a reduced number of open stacks during the cutting process because if the number of stacks increases beyond the available space, stacks must be removed in order to give space to the new stacks. Closed stacks can be removed definitively to another place or can be delivered to clients but, open stacks, if removed, must be brought back later to be completed when the corresponding item type is cut again in another pattern. This is not efficient and consumes time and resources. Therefore, there is an interest in determining a sequence to cut the patterns

that minimizes the maximum number of open stacks during the cutting process. This sequencing problem is known as the minimization of open stacks problem (MOSP).

Formally we can define MOSP as follows: given a Boolean matrix C in which the columns correspond to the patterns and each row corresponds to the item types, the entry c_{ij} is equal to 1 if and only if item i is contained in pattern j (the number of items i in pattern j is not relevant). Consider C_π , the resulting matrix obtained by a permutation π of the columns of C and let C_π^1 be the corresponding matrix obtained from C_π that has the consecutive 1's properties, that is, in any row of this matrix, all elements between two elements 1 is also 1. The objective is to find a permutation π of the columns, such that the maximum number of 1's in any column of matrix C_π^1 is minimized.

Recently, MOSP was focused in the Constraint Modeling Challenge 2005 (see Smith and Gent, 2005) in the Fifth Workshop on Modelling and Solving Problems with Constraints. This challenge increased the interest on the MOSP by the Operations Research scientific community.

The literature on the MOSP is not extensive. Lins (1989) considered a particular case of MOSP where there are at most 2 different item types per pattern. Considering these simple patterns he built a graph associating nodes to items and arcs to patterns and obtained 1-tree graphs (i.e., a tree with an additional arc) for his practical instances. He proposed a heuristic method to solve these instances. Yuen (1991, 1995) presented six simple heuristics for MOSP. The first two heuristics use two steps, in the first, an open item is selected and in the second a pattern that contains the selected item is sequenced. In the other four heuristics the selection

* Corresponding author. Tel.: +55 12 39456551; fax: +55 12 39456375.

E-mail addresses: horacio@lac.inpe.br (H.H. Yanasse), elfsenne@fegunesp.br (E.L.F. Senne).

of the next pattern to sequence is based on a measure of “matching” between a pattern and the current open items. Yuen’s heuristic number 3 was considered the most efficient by the computational tests performed by the author. Yuen and Richardson (1995) presented a trivial lower bound for MOSP given by the maximum number of different item types in the patterns. They proposed the decomposition of the problem by identifying clusters of patterns (see pre-processing 1, reviewed in Section 2) and an exact method that enumerates permutations of pattern sequences. Upper bounds given by the heuristics proposed in Yuen (1991, 1995) and the new lower bound proposed are used to reduce the search space. Yanasse (1996) proposed polynomial algorithms for the 1-tree case and other graphs with very special topologies. Yanasse (1997a) presented an exact polynomial algorithm for the tree case. Yanasse (1997b) propose a non-trivial mathematical formulation for MOSP, a branch and bound algorithm and a greedy heuristic algorithm. He also presents relations of the MOSP with the Minimization of Order Spread Problem (see Garey et al., 1978; Madsen, 1988), and the minimization of the number of tool switches problem (see Tang and Denardo, 1988). Limeira (1998) and Yanasse and Limeira (1998, 2004) integrated some polynomial algorithms developed to solve special classes of instances of MOSP with the branch-and-bound algorithm proposed by Yanasse (1997b). Yanasse et al. (1998) proposed a polynomial algorithm to solve another special instance of MOSP, where the MOSP graph is a complete graph with tree subgraphs attached to its nodes. Yanasse (1997c) showed that any instance of MOSP, not only those with at most two item types per pattern as in Lins (1989), can be modeled as a graph problem. The MOSP graph is obtained as follows. Each node in the graph corresponds to an item type and there is an arc connecting two nodes if and only if there is at least one pattern containing both items types corresponding to the nodes. Therefore, a pattern with k item types is represented by a clique of size k in the MOSP graph. Observe that the MOSP graph is a union of cliques and it is completely different from the pattern graph connection suggested in Yuen and Richardson (1995). Yanasse (1997c) defined the MOSP as a graph problem as the problem of traversing the arcs of the graph such that the number of simultaneously open nodes is minimized. A node is considered closed when all the arcs incident to it are traversed, a node is considered open after the first arc incident to it is traversed. Yanasse (1997c) showed then that any MOSP instances that produce the same MOSP graph are solution-wise equivalent. In addition, since any MOSP graph can be generated by an instance of MOSP where all patterns have at most two item types, solving this particular class of MOSP instances is computationally not easier than solving any other instance of MOSP. Yanasse (1997c) also introduced two new lower bounds for MOSP given by the size of the maximal clique of the MOSP graph and the smallest degree of any node in the MOSP graph. He also showed that the maximum number of open stacks following a given sequence of patterns and its reverse sequence is the same; any MOSP having more than one pattern to sequence has multiple optimal solutions; given two instances 1 and 2 of MOSP, if 1 has a subset of the patterns of 2, then the optimal value of instance 2 is greater or equal to the optimal value of instance 1. With this last result, lower bounds for an instance of MOSP can be obtained by solving smaller MOSP instances, for instance, lower bounds can be obtained using subgraphs of the original MOSP graph. Yanasse (1997c) also showed how to obtain an optimal solution to the original problem given an optimal solution of the MOSP as a graph problem. The solution is obtained by observing, just after each arc is traversed, all the open nodes. Following the optimal sequence to traverse the arcs, the first time that all nodes (items) of a pattern of the original problem are open or closed, that pattern is sequenced. The pseudo-code of this algorithm is presented next.

Algorithm to determine an optimal solution of the original problem given an optimal solution of the MOSP graph problem

INPUT: Patterns of the original problem; Optimal sequence to traverse the arcs of the MOSP graph.

OUTPUT: Corresponding optimal sequence of the patterns.

1. S = the set of original patterns; $L(\text{arcs})$ = ordered list of arcs of the MOSP graph to traverse; $L(\text{patterns})$ = empty; OPEN = empty; CLOSED = empty.
2. While S is not empty do:
 - 2.1. Get next arc of $L(\text{arcs})$;
 - 2.2. Update OPEN with the open nodes and CLOSED with the closed nodes;
 - 2.3. Check for all patterns in S that have all their item types in set OPEN or in set CLOSED . If any, include the pattern in $L(\text{patterns})$ and delete it from S .
- Endwhile;
3. Return $L(\text{patterns})$.

The computational complexity of this algorithm is $O(m^2 n \log m)$, where m is the total number of item types and n is total number of patterns.

Faggioli and Bentivoglio (1998) presented a mathematical model for MOSP and a three phase solution method. In the first phase, a good solution is obtained with a greedy heuristic similar to some of Yuen’s (1995) heuristics. In the second phase this solution is improved using tabu search; and in the third phase, an implicit enumeration scheme of permutation of patterns is used that is just an improved procedure of Yuen and Richardson (1995). Faggioli and Bentivoglio (1998) ignore the dominated patterns in the enumeration since this does not lose the optimal solution. This is, in fact, the pre-processing operation suggested in Becceneri (1999) and in Becceneri et al. (2004) that is reviewed later in Section 2 of this manuscript, pre-processing 2.

In Yanasse et al. (1997, 1999) and Becceneri (1999), an arc contraction heuristic is presented. This arc contraction operation is also used to develop a new lower bound to the optimal value of MOSP in Yanasse et al. (1999). To the best of our knowledge, this arc contraction lower bound dominates all previous lower bounds proposed in the literature. In Becceneri (1999), the least cost node heuristic for MOSP was proposed and it was later modified in Becceneri et al. (2004). In this heuristic, the next arcs of the MOSP graph to be traversed are chosen by closing the node with the least number of arcs incident to it. To the best of our knowledge, the performance of this heuristic, in terms of the quality of the solution, is the best or among the best of the literature. Ashikaga (2001) presented two new heuristics for solving the MOSP. In the first he searches for Hamiltonian circuits and for that, he uses a variation of the extension–rotation algorithm of Pósa (1976). This algorithm starts with a path with a single node and tries to extend it as far as possible. Ashikaga’s (2001) variation consisted in choosing the set of nodes of the maximal clique of the MOSP graph, instead of a single node and to restrict the extensions. The maximal clique is determined by a greedy search for the minimum grade vertex of the complementary MOSP graph. The second heuristic proposed by Ashikaga (2001), uses a recursive contraction of maximal cliques until a single hyper-node is obtained. By the computational tests performed, Ashikaga’s first heuristic was better than Yuen’s heuristic 3, in terms of quality of solution obtained and smaller execution times. Meta-heuristics were also suggested for solving the MOSP. Linhares et al. (1999) developed a simulated annealing heuristic, Fink and Voss (1999) used simulated annealing and tabu search, Oliveira and Lorena (2002a,b) used a constructive genetic algorithm. It is worth observing that Linhares (2001) and Oliveira and Lorena (2002a) reported better solutions for some data set of instances compared to the solutions presented by Faggioli and

Download English Version:

<https://daneshyari.com/en/article/478675>

Download Persian Version:

<https://daneshyari.com/article/478675>

[Daneshyari.com](https://daneshyari.com)