Computers & Operations Research 40 (2013) 991–1003

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

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

Heuristics for the strip packing problem with unloading constraints ☆

Jefferson L.M. da Silveira *, Flávio K. Miyazawa, Eduardo C. Xavier

Institute of Computing, University of Campinas – UNICAMP, Campinas, SP, Brazil

ARTICLE INFO

Available online 22 November 2012

Keywords:
Strip Packing
GRASP
Approximation algorithms

ABSTRACT

This article addresses the Strip Packing Problem with Unloading Constraints (SPU). In this problem, we are given a strip of fixed width and unbounded height, and n items of C different classes. As in the well-known two-dimensional Strip Packing problem, we have to pack all items minimizing the used height, but now we have the additional constraint that items of higher classes cannot block the way out of lower classes items. This problem appears as a sub-problem in the *Two-Dimensional Loading Capacitated Vehicle Routing Problem* (2L-CVRP), where one has to optimize the delivery of goods, demanded by a set of clients, that are transported by a fleet of vehicles of limited capacity based at a central depot. We propose two approximation algorithms and a GRASP heuristic for the SPU problem and provide an extensive computational experiment with these algorithms using well know instances for the 2L-CVRP problem as well as new instances adapted from the Strip Packing problem.

1. Introduction

In recent years some attention has been devoted to the combination of two problems: the two-dimensional packing and the routing problem. The combination of these two problems models situations where one aims to deliver goods, demanded by customers, that are transported by vehicles of limited capacity based at a central depot. This problem is called *Two-Dimensional Loading Capacitated Vehicle Routing Problem* (2L-CVRP) [25]. The objective is to generate a set of routes of minimum total cost that covers all clients, where each route induces feasible packings, i.e., all items of one route must be packed in one vehicle satisfying the traditional packing constraints and a new unloading constraint. The unloading constraint is the following: given a set of items that are delivered along a route, while delivering items of one client, there must not exist items of other clients ahead on the route blocking the way out of the items of the current client.

One important task in algorithms for the 2L-CVRP problem (see [20,18,26,15]) is to check if a giving route induces a valid packing. One way of doing this is solving a strip packing problem with the unloading constraint and checking whether the generated packing height is smaller or larger than the maximum allowable height. In this work we focused on this problem called here by Strip Packing with Unloading Constraints (SPU).

We can define the SPU problem as follows: given a strip S of fixed width and unbounded height, and a list of items of C different classes, each item a_i of height $h(a_i)$, width $w(a_i)$ and class $c(a_i)$, we must pack the items into S minimizing the used height. Furthermore, if an item a_i has class greater than a_j, i.e. $c(a_i) > c(a_j)$, then a_i must not block the way out when removing item a_j. We also consider the case in which 90° rotations are allowed (SPU^r). This problem is strongly NP-Hard since it is a generalization of the Two-dimensional Strip Packing problem.

Papers which addresses the 2L-CVRP problem used some simple heuristics or exact algorithms to tackle the packing problem, and do not provide information about the quality of the solutions (except [18] that presented the average occupied area in the vehicles (bins) of the problem).

In [20] Iori et al. proposed an exact algorithm to the 2L-CVRP problem. Their packing algorithm is the bottom-left heuristic and a branch-and-bound procedure to check the feasibility of the loadings. Their solution can solve instances involving up to 25 clients and 91 items in 1 day of CPU time.

Gendreau et al. [18] proposed a tabu search algorithm to the 2L-CVRP problem. The packing problem is solved using heuristics, local search and a truncated branch-and-bound. Their algorithm iteratively applies a procedure based on the Touching Perimeter algorithm [27] for the two-dimensional bin packing problem (it is worth noting that the Touching Perimeter heuristic is also used in the two-dimensional strip packing problem [23]). At first, the items are sorted in reverse order of clients visit, and a packing is constructed. Subsequently the algorithm tries to improve the packing perturbing the trivial order that items were packed.

In [26] Kiranoudis et al. proposed a guided tabu search heuristic to the 2L-CVRP. They used five different heuristics (in

* Corresponding author.
E-mail addresses: jmoises@ic.unicamp.br (J.L.M. da Silveira), fkm@ic.unicamp.br (F.K. Miyazawa), eduardo@ic.unicamp.br (E.C. Xavier).

☆ This research was supported by CNPQ and FAPESP.

0305-0548/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2012.11.003

order) to tackle the packing problem. The first and second heuristics are based on the bottom-left heuristic. The third and fourth heuristics are similar to the one used by Gendreau et al. [18], based on the Touching Perimeter.

The best result for the 2L-CVRP is due to Doerner et al. [15] and their *Ant Colony Optimization* heuristic. The authors use *Bin-packing* lower bounds to prove unfeasibility of some routes, and then heuristics to construct the packing solutions. Their heuristics are quite similar to the ones previously cited in [18,26] (Bottom-left and Touching Perimeter) and also use a truncated branch-and-bound with a limited CPU time.

One of the best results for the Strip Packing problem without rotations was obtained by a Reactive GRASP heuristic proposed by Alvarez et al. [1]. Some other papers also achieved similar results [5,11,8].

In [3], Azar and Epstein proposed an online 4-competitive algorithm to a version of the strip packing problem, where while packing one item there must be a free way from the top of the bin until the position where the item is packed. In this model, a rectangle arrives from the top of S, and it should be moved continuously using only the free space until it reaches its place, as in the well known *TETRIS game*. Their online algorithm can be easily modified to an *offline* algorithm to the relaxed version of the SPU problem where the items can use vertical and horizontal movements to leave the bin.

Fekete et al. [16], proposed an online 2.6154-competitive algorithm to a version of the square strip packing problem, similar to the one considered in [3]. In this algorithm, the items are packed from the top of S and are moved only with vertical movements to reach its final position. In addition, an item is not allowed to move upwards and has to be supported from below when reaching its final position. These conditions are called *gravity constraints*. Their slot based algorithm can be easily used to the SPU problem, achieving an 2.6154-approximation, in the special case where items are squares. We just need to sort the items in non-increasing order of class values.

Finally, Augustine et al. [2] present approximation algorithms forr a related problem. They consider the strip packing problem with precedence constraints and/or with release dates. Their problem has applications in scheduling problems for FPGA.

1.1. Our results

For the SPU^r problem, we propose a bin packing based heuristic and prove that this heuristic is a 6.75-approximation algorithm. Besides that, we also propose an 1.75-approximation algorithm for a special case of the SPU problem, where the number of classes (clients in a route) is bounded by a constant. This algorithm is based on the well known *First-Fit-Decreasing Height* algorithm [14].

Finally, we propose a GRASP heuristic for the SPU problem that is based on the Reactive GRASP heuristic presented in [1]. We adapted this heuristic to consider the unloading constraint and also for the SPU^r problem. We changed the focus of the algorithm to the items classes instead of their dimensions.

Besides the theoretical results presented for the approximation algorithms, their practical performance is also checked. The effectiveness of the proposed heuristics is demonstrated through extensive computational experiments on benchmark instances [30]. We also generated several new instances based on benchmark instances for the strip packing problem [31,4,6,9,12,21,22,7].

We show that our algorithms achieve a good occupation of the area of the strip in low CPU time. We also show that our best packing heuristics improves the solutions of the 2L-CVRP problem when compared to other well know heuristic.

1.2. Paper organization

This paper is organized as follows: in Section 2 we introduce our definitions and formalize the description of the SPU problem. The approximation algorithms are presented in Section 3. In Section 3.1 we present an asymptotic 6.75-approximation algorithm for the SPU^r problem and in Section 3.2 we present an asymptotic 1.75-approximation algorithm for the special case of the SPU problem, where the number of classes in an instance is bounded by a constant. In Section 4 we present the constructive algorithm and the Local Search strategy used in the GRASP based heuristics which are described in Section 5. In Section 6 we present the instances used on the experiments. In Section 7 we summarize our computational experiments and results. Moreover, we present lower bounds used in this work. Finally, in Section 8 we analyze the results and argue about the effectiveness of the proposed heuristics and approximation algorithms.

2. Definitions and notation

We define the SPU problem as follows: an instance of the problem is composed by a strip S of fixed width W and unbounded height, and a list L of n items, each item a_i with height $h(a_i)$, width $w(a_i)$ and class $c(a_i)$. The class values $c(a_i)$ are interpreted as an order of removal of the item from the strip. A packing is defined by a function that maps each item a_i to a point $(x(a_i),y(a_i))$, where $x(a_i)$ and $y(a_i)$ are the coordinates of the bottom-left corner of the item a_i on S. The bottom-left corner of the strip has coordinates (0,0). The goal is to pack all items into S minimizing $\max_i\{y(a_i)+h(a_i)\}$, $1 \le i \le n$, subject to the constraints:

- All the items must be completely contained in S.
- Items can not overlap each other.
- All the items must satisfy the unloading constraint (see Fig. 1), i.e., for any two items $a_i,a_j \in L$, where $c(a_i) > c(a_j)$, we must have $x(a_i)+w(a_i) \le x(a_j)$ or $x(a_j)+w(a_j) \le x(a_i)$ or $y(a_i)+h(a_i) \le y(a_j)$. This imposes that each item can be removed from the strip in increasing order of classes using only vertical movements.

Let $\mathcal{A}$ be an algorithm for the SPU problem and let $\mathcal{A}(I)$ be the cost of the solution computed by $\mathcal{A}$ for instance I. We say that $\mathcal{A}$ is an α-approximation algorithm if it has polynomial time complexity, and for every I it satisfies $\mathcal{A}(I) \le \alpha \mathrm{OPT}(I)$, where $\mathrm{OPT}(I)$ is the cost of an optimum solution to instance I. As it is common in packing problems, we consider in this work asymptotic approximation algorithms, where in this case the algorithm must satisfy $\mathcal{A}(I) \le \alpha \mathrm{OPT}(I)+\beta$ for some constant β (Fig. 1).

3. Approximation algorithms

3.1. A 6.75-approximation algorithm for the SPU^r problem

In this section we present the *Hybrid Bin Packing* (HBP) algorithm to solve the SPU^r problem. Without loss of generality, we assume that the width of the strip is 1 and all items have width and height at most 1. The HPB algorithm computes the solution in two stages. The algorithm uses in the first stage, a bin packing algorithm, which we call *Level Bin Packing* (LBP). This bin packing algorithm packs the items into bins (rectangles) of height 1 and width 1, using levels or, what we call, horizontal sub-strips (horizontal slices of a bin, see Fig. 2). Then the HBP algorithm, using the bins computed by the LBP algorithm, concatenates these bins in such a way to obtain a feasible strip S. Due to the form that