Summary of Master's Thesis

-Bikash Adhikari

An LP bound for 2-Dimensional Knapsack by 1-Dimensional Cutting Stock Relaxation

In industrial processes as well as logistics applications, it is often required to allocate small rectangular items to larger rectangular items with a minimum waste. For example, in textile or paper manufacturing, standardized roll of cloths or cardboard are given and one needs to obtain certain required items by using a minimum roll length. In many applications, standard stock units are generally large rectangular sheets of one or few different sizes and the goal is to minimize the total area of the sheets necessary to produce given amounts of small rectangles of given sizes.

In literature, the relevant optimization models are referred to as two-dimensional (2D) BIN PACKING or CUTTING STOCK problems, the difference lying in the number of each small item required (one or few units in the former case and many in the latter). In fact, those models extends well beyond stock cutting applications, and include task scheduling, VLSI design, image processing, packing goods in crates, commercials assignment to TV breaks, truck loading, just to name a few.

A very important problem both if considered stand-alone and when arising from Dantzig-Wolfe decomposition of 2D BIN PACKING aims at maximizing the total value of the items packed into a single bin of given size. Namely, consider n rectangular items R_i, $i = 1, 2, \cdots, n$, of width w_i and height h_i, each associated with a real positive value p_i to be packed into a large stock rectangle R_0 of width W and height H (the bin). Without loss of generality, assume both edges of each R_i specified as positive integers, $i = 0, 1, \cdots, n$, and $w_i \leq W$, $h_i \leq H$ for $i = 0, 1, \cdots, n$. Assume also that the R_i's must be packed into R_0 with their edges parallel to those of R_0 and rotation (that is, interchanging w_i and h_i) is not allowed. We want to

Pack items into the bin such that the total value of the items packed is maximum.

This optimization problem is known as 2D 01 KNAPSACK and, as a generalization of ordinary (1-dimensional) 01 KNAPSACK is recognized NP-Hard (cite here). This implies that a polynomial time algorithm for this problem cannot be found unless $P = NP$ or, saying it differently, that the best algorithm known so far to find an optimal solution requires, in the worst case, a number of steps exponential in the size of the problem.

Due to problem complexity, finding good upper and lower bounds to the value of optimal solutions can therefore be crucial in order to improve the practical efficiency of exact solution algorithms. In this work, we study upper bounds obtained by relaxing 2D 01 KNAPSACK into a special 1D CUTTING STOCK problem. The intuition under this relaxation is easily explained: we imagine to cut the 2D items along the width into pieces of unit width, and do the same with the bin: as a result, the i-th item generates w_i 1-dimensional pieces of height h_i, and the bin generates W pieces (regarded as 1D stock items) of height $H \geq h_i$. We then try to obtain small pieces for a total maximum value by solving a modified 1D CUTTING STOCK problem where, unlike the classical case, only W stock items are available and item i, if cut from stock, must be produced in the demanded amount w_i (from now on, we reserve the term *item* to 2D parts and the term *piece* to 1D ones).

The described relaxation does not remove inequalities from a particular formulation of 2D 01 KNAPSACK, but rather weakens a geometrical properties of items (viewed as real 2D intervals), that is connectivity. A further relaxation is done by removing integrality constraints in the 1D CUTTING STOCK problem. This removal simplifies computation, as the linear relaxation can be computed by column generation.

Column generation is successfully employed to compute very good linear relaxations of CUTTING STOCK. The methods aims at solving a *Master Problem (MP)* of the form

$$(MP) \qquad \begin{aligned} z^*_{MP} = \min \quad & \sum_{j \in J} x_j \\ s.t. \quad & \sum_{j \in J} a_{ij} x_j \geq d_i \qquad \forall i \in I \\ & x_j \geq 0, \qquad \forall j \in J \end{aligned} \tag{1}$$

where J is the set of patterns, x_j is the number of times pattern $j \in J$ is used, a_{ij} is the number of pieces of type i obtained from pattern j and d_i is the demand for each piece i. This linear program has $|J| = N$

variables and $|I| = n$ constraints notice that N is exponential in n because feasible patterns are in one-to-one correspondence with the solution of a complex packing problem (e.g., 01 KNAPSACK). Therefore, the master problem cannot be solved as is, and we consider the *Restricted Master Problem (RMP)* which contains a limited subset J' of patterns (with the only initial condition that every item i has a non-zero entry in at least one pattern $j \in J'$):

$$
\begin{aligned}
& & z^*_{RMP} = \min \quad & \sum_{j \in J'} x_j & \\
(RMP) \qquad & s.t. & \sum_{j \in J'} a_{ij} x_j \geq d_i \qquad & \forall i \in I & \qquad (2)\\
& & x_j \geq 0, \qquad & \forall j \in J' &
\end{aligned}
$$

Promising patterns that is, columns entering the Simplex basis are selected by implicit enumeration, minimizing reduced costs via the solution of a pricing problem. Let π be the dual variables of the *RMP*, whose dual reads

$$
\begin{aligned}
& \max \quad \sum_{j \in J} \pi_i d_i & \\
s.t. \qquad & \sum_{j \in J} \pi_i a_{ij} \leq 1 \qquad \forall i \in I & \qquad (3)\\
& \pi_i \geq 0, \qquad \forall j \in J &
\end{aligned}
$$

and let π^* be an optimal dual solution. In order to find (if any) a column (primal variable) with negative reduced cost it is necessary to solve a *Pricing Problem (PP)*. In 1D CUTTING STOCK, the pricing problem takes the form of 1D INTEGER KNAPSACK:

$$
\begin{aligned}
& \max \quad & \sum_{i \in I} \pi^*_i y_i & \\
(PP) \qquad & s.t. \quad & \sum_{i \in I} \ell_i y_i \leq L & \qquad (4)\\
& & y_i \in \mathbb{Z}^+ &
\end{aligned}
$$

where ℓ_i is the length of the piece i and L is the length of the stock from which demanded pieces are cut. A solution y of the *PP* gives a new column of the *RMP*, and the resulting problem is re-solved by a pivot step. The new dual optimum defines a new *PP*, and the process is repeated until no column with negative reduced cost is found.

In our case, an upper bound to 2D 01 KNAPSACK is computed by choosing a limited set of W patterns so that the demand of each piece is fully satisfied or the piece is not produced at all, and the total value of the pieces produced is maximized. Such a bound is the optimum value of the integer linear program:

$$
\begin{aligned}
& \max \quad \sum_i \pi_i y_i & \\
s.t. \qquad & \sum_k p_{ik} x_k = w_i y_i \qquad i = 1, 2, \cdots, n & \\
& \sum_k x_k = W & \qquad (5)\\
& y_i \in \{0, 1\} \qquad i = 1, 2, \cdots, n & \\
& x_k \geq 0, \qquad \text{integer} &
\end{aligned}
$$

where $y_i = 1$ if item i is produced and 0 otherwise, x_k is the run length of pattern k, $p_{ik} \in \{0, 1\}$ is the number of units of piece i produced by pattern k, w_i is the demand of each piece of type i (= item width, or the number of strips in which item i is sliced), and π_i is the value of item i. In the above program, the first set of equations fulfills the demand of piece i whenever $y_i = 1$, the last equation ensures that the amount of patterns chosen equals the width of the stock rectangle. Note that a second bound is symmetrically obtained by replacing W, w_i by H, h_i: one then chooses the best (i.e., the least) of the two.

Problem (5) can be solved by *Branch-and-price*. A weaker (but easier) bound is however obtained by LP relaxation of (5). We implemented the relaxation using PYTHON and GUROBI. A bound computed in this way has been evaluated and compared to optimum 2D packings for several problem instances found in the literature.

The full text of Master's thesis is available here. Masters Thesis Full