Universität Hamburg

Università Degli Studi Dell'Aquila

Universitat Autònoma de Barcelona

Joint MSc Programme in
Mathematical Modelling in Engineering: Theory, Numerics, Applications

Master's Thesis

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

Supervisor
Prof. Claudio Arbib
University of L'Aquila, Italy

Candidate
Bikash Adhikari
Matricola : 249542

Table of Contents

Table of Contents

Two-Dimensional Knapsack Problem

- **Input:** n rectangular items $i \in I = \{1, 2, \ldots, n\}$ of width w_i, height h_i, value c_i; a big rectangular bin of size (W, H).
- **Problem:** Pack the items into the knapsack so that the total value of the items packed is maximized.
- **Application:** Logistics, cutting processes in industries (paper, rubber, textile etc.), task scheduling, commercial assignments to TV breaks etc.
- **Complexity:** NP-Hard[1].
- **Goal:** Determine good upper bounds in an efficient way (crucial for exact solution algorithms).
- **Approach:** Relaxation of 2D Knapsack into 1D Cutting Stock problem.

[1] Michael R. Garey and David S. Johnson. Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA, 1990.

One-Dimensional Cutting Stock Problem

- **Input:** Set I of *small unidimensional pieces* of height h_i; piece $i \in I$ required in w_i copies; integer W = number of *big pieces* available at stock.
- **Problem:** Cut the W stock pieces according to *patterns* (1D Knapsack solutions) so as to produce w_i copies of piece $i \in I$.
- **Application:** The 1D Cutting Stock problem (1CSP) formulates cutting process optimization in such industries as steel, aluminum, rubber, paper etc.
- **Formulation:** Best results with Dantzig-Wolfe decomposition + branch-and-price (linear integer program with one integer variable per pattern). Exponentially many patterns (1D Knapsack solutions) to be managed by delayed column generation.
- **Complexity:** NP-hard[2].

[2] Michael R. Garey and David S. Johnson. Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA, 1990.

Nature of Cutting and Packing

- Share the same nature and the only difference between them is the perspective from which the problem is investigated.
- Have we had several items with the same weights, Packing distinguishes among them and appends a label to each item.
- Cutting Stock associates all the items of the same weight as required demand.
- Either formulation leads to the same optimal solution.
- Column generation is successfully employed to compute very good linear relaxations of Cutting Stock problems.

Column Generation

- Column Generation[3] is used to solve the linear relaxation by iterative addition of variables (columns of the LP).

- **Idea:** Initialize and update a *Restricted Master Problem* (RMP) with a subset $J' \subseteq J$ of columns:

$$\min \quad \sum_{j \in J'} c_j x_j \tag{1}$$

$$\text{s.t.} \quad \sum_{j \in J'} a_{ij} x_j \geq b_i \qquad \forall i \in I \tag{2}$$

$$x_j \geq 0, \qquad j \in J'. \tag{3}$$

- Solve RMP to optimality. The optimal value z^*_{RMP} of the objective function is an upper bound to the MP optimum: $z^*_{MP} \leq z^*_{RMP}$.

[3] R C E Gilmore and Ralph Gomory. A linear programming approach to the cutting stock problem i. 9, 01 1961.

Pricing Problem

- To close the gap $z^*_{RMP} - z^*_{MP}$ one has to find a variable candidate to enter the current basis, i.e., one with negative reduced cost: if no such variable exists, the current solution is MP-optimal.
- Let π_i be the dual RMP variables and π^* a dual RMP optimal solution. to find (if any) a variable with negative reduced cost, solve an optimization problem called *pricing*:

$$\bar{c} = \min_{j \in J} \left\{ c_j - \sum_{i \in I} \pi^*_i a_{ij} \right\}. \tag{4}$$

FlowChart of Column Generation Algorithm

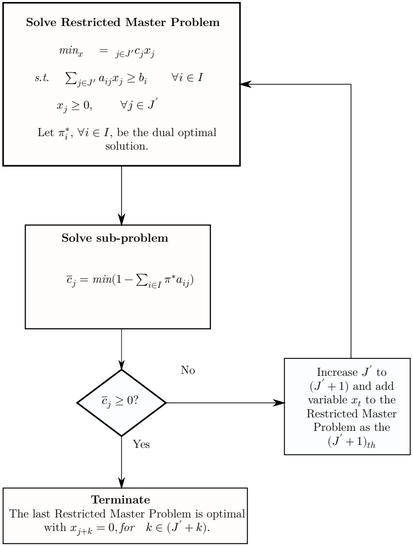


Figure: Flowchart: Column Generation

Table of Contents

Outline of Relaxation

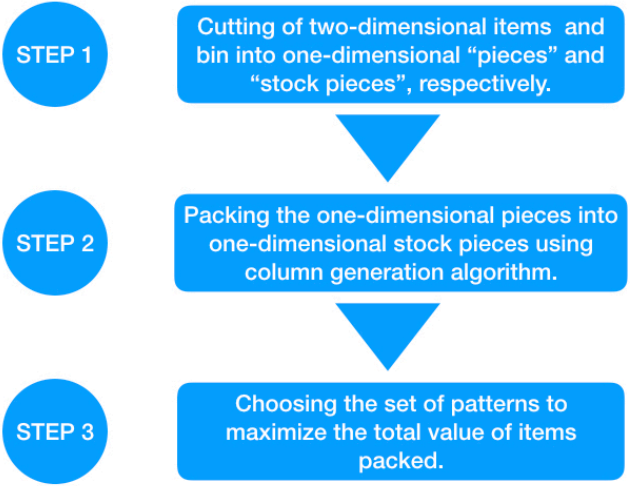
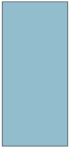


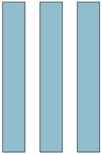
Figure: Outline of the relaxation methodology.

Relaxation of 2D Knapsack

- *Idea:* Relax connectivity by cutting both the 2D items and the bin along the height into pieces of unit width.
- Slicing item $i \in I$ generates w_i 1D small pieces of height h_i (item values uniformly divided among pieces).
- Slicing the bin generates W stock pieces of height $H \geq h_i$.
- Solve the resulting 1D Cutting Stock problem with W stock items available.
- Further relaxation done by removing integrality constraints of 1CSP.



2D item with $w = 3$, $h = 7$ and $c = 35$



1D pieces obtained after slicing the 2D item, each with $w_i = 1$, $h_i = 7$ and $c_i = 11.66$

Master Problem

- We formulate following linear program to pack the pieces into stock pieces.

$$
\begin{aligned}
& & min \quad & \sum_{j \in J} x_j & \\
& s.t. & & \sum_{j \in J} p_{ij} x_j \geq w_i & \forall i \in I \\
& & & x_j \geq 0, & \forall j \in J
\end{aligned}
\tag{5}
$$

where J is the set of patterns, x_j is the number of times pattern $j \in J$ is used, p_{ij} is the number of pieces of type i obtained from pattern j and w_i is the demand for each piece i.

- $\mid J \mid = N$ is exponential in $\mid I \mid = n$ because feasible patterns are in one-to-one correspondence with the solution of packing problem.

- The master problem cannot be solved as it is, he we consider the restricted master problem which contains limited subset of $J' \subseteq J$ of patterns.

Restricted Master Problem

- Easy start will be to consider $\mid I \mid$ single-pieces cutting patterns.

$$\begin{aligned} & min \quad \sum_{j \in J'} x_j \\ s.t. \quad & \sum_{j \in J'} p_{ij} x_j \geq w_i \qquad \forall i \in I \\ & x_j \geq 0, \qquad \forall j \in J' \end{aligned} \tag{6}$$

- Solving the above RMP we obtain primal optimal solution x^* and dual optimal solution π^*.
- Promising patterns – that is, columns entering the current basis – are selected by implicit enumeration, minimizing reduced costs via the solution of a pricing problem.

Pricing Problem

- In order to find (if any) a column (primal variable) with negative reduced cost it is necessary to solve a Pricing Problem (PP) which takes the following form:

$$\begin{aligned} max \quad & \sum_{i \in I} \pi_i^* z_i \\ s.t. \quad & \sum_{i \in I} h_i z_i \leq H \\ & z_i \leq UB_i \quad \forall i \in I \\ & z_i \in \mathbb{Z}^+ \quad \forall i \in I \end{aligned} \tag{7}$$

  where h_i is the height of the piece i and H is the height of the stock from which demanded pieces are cut and UB_i is the maximum number of times each item is allowed to be packed into the bin.

- If $\sum_{i \in I} \pi_i^* z_i^* \leq 1$ then x^* is the optimal solution.

- If not, then the corresponding column is add to the RMP and the process is repeated until no column with negative reduced cost is found.

Integer Linear Program

- Upper bound: Computed by choosing a limited set of W patterns so that the demand of each piece is fully satisfied and the total value of the pieces produced is maximized.

$$
\begin{aligned}
& max && \sum_i c_i y_i \\
s.t. \quad & && \sum_k p_{ik} x_k = w_i y_i \qquad i = 1, 2, \cdots, n \\
& && \sum_k x_k = W \\
& && y_i \le UB_i \qquad \forall i \\
& && y_i \ge 0, \qquad \text{integer} \\
& && x_k \ge 0, \qquad \text{integer}
\end{aligned}
\tag{8}
$$

where $y_i \in \mathbb{Z}^+$ is the number of item i produced, x_k is the run length of pattern k, p_{ik} 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 c_i is the value of item i.

Integer Linear Program

- In the above program, the first set of equations fulfills the demand of piece i , the second constraint ensures that the amount of patterns chosen equals the width of the stock rectangle and the last one ensure that the number of items packed is always less than the upper bound UB.
- We solve the LP relaxation of the above program to obtain the required bounds.
- The second bound is symmetrically obtained by replacing W, w_i by H, $h_i$and we choose the best (i.e., the least) of the two.

Example

$Items$	$Size(w,h)$	LB	UB	$Value(c)$
$Item1$	$(3,7)$	0	2	35
$Item2$	$(8,2)$	0	2	40
$Item3$	$(10,2)$	0	1	27
$Item4$	$(5,4)$	0	3	23
$Item5$	$(2,9)$	0	2	43

Table: Set of 2D items that should be packed into the bin of size (10,10): size, minimum and maximum number of each item that is allowed to pack into the bin and their respective value.

$Pieces$	$Size$	$Demand(w)$	c_p
$Piece1$	$(1,7)$	3	11.66
$Piece2$	$(1,2)$	8	5
$Piece3$	$(1,2)$	10	2.7
$Piece4$	$(1,4)$	5	4.6
$Piece5$	$(1,9)$	2	21.5

Table: Set of the 1D pieces obtained after slicing 2D items.

Initial Sets of Patterns for solving RMP

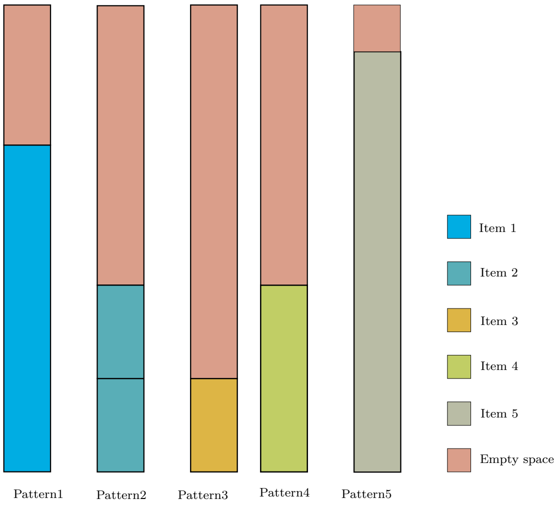


Figure: Initial set of patterns for solving the RMP

Patterns Generated by Column Generation Algorithm

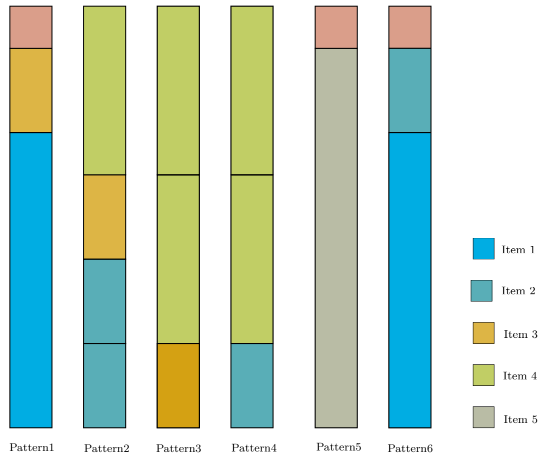
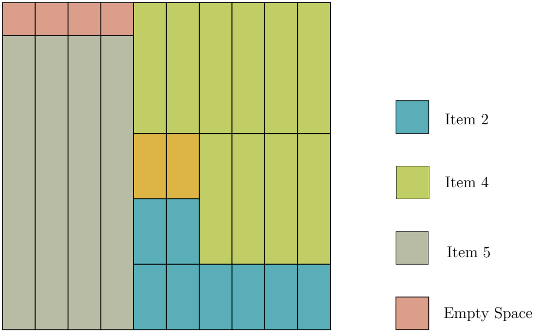


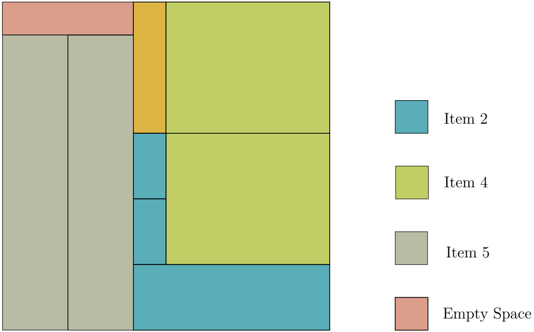
Figure: Patterns generated by column generation algorithm.

Stages of 2D Knapsack packing by 1D Relaxation



- From the solution of ILP, we have $x_1 = 0$, $x_2 = 2$, $x_3 = 0$, $x_4 = 4$ and $x_5 = 4$ and $y_1 = 0$, $y_2 = 1$, $y_3 = 0$, $y_4 = 2$ and $y_5 = 2$.

Stages of 2D Knapsack packing by 1D Relaxation



- Hence the optimal value is : $c_1 * y_1 + c_2 * y_2 + c_3 * y_3 + c_4 * y_4 + c_5 * y_5 =$ 35*0 + 40*1 + 27*0 + 23*2 + 43*2 = 0 + 40+ 0 + 46 + 86 = 172.

Computational Reference

- **Implementation:** PYTHON and GUROBI.
- **Data Sets:** OR-LIBRARY[4].
- The results obtained are compared with the one provided in J. E. Beasely, 1985[5].
- Upper bounds in the literature were obtained using Lagrangean relaxation and further minimized using subgradient optimization.

[4] http://people.brunel.ac.uk/ mastjjb/jeb/info.html

[5] J. E. Beasley, (1985) An Exact Two-Dimensional Non-Guillotine Cutting Tree Search Procedure. Operations Research 33(1):49-64. http://dx.doi.org/10.1287/opre.33.1.49

Computational Results

No.	*Bin Size*	*No. of Items*	*UB*	*OPT*	UB_{ver}	*Time (Sec)*	UB_{hor}	*Time(Sec)*	UB_{best}
1	(10,10)	5	164	164	172	0.05	201	0.03	172
2	(10,10)	7	247	230	250	0.04	232	0.06	232
3	(10,10)	10	260	247	254	0.04	253	0.04	253
4	(15,10)	5	268	268	268	0.03	268	0.03	268
5	(15,10)	7	358	358	373	0.04	358	0.05	358
6	(15,10)	10	317	289	317	0.03	291	0.07	291
7	(20,20)	5	430	430	430	0.01	430	0.03	430
8	(20,20)	7	915	834	897	0.05	938	0.03	897
9	(20,20)	10	930	924	955	0.05	953	0.05	953
10	(30,30)	5	1452	1452	1452	0.03	1517	0.05	1452
11	(30,30)	7	1860	1688	1737	0.05	1864	0.04	1737
12	(30,30)	10	1982	1865	1875	0.09	1983	0.06	1875

Table: Computation Results

Bound Comparision

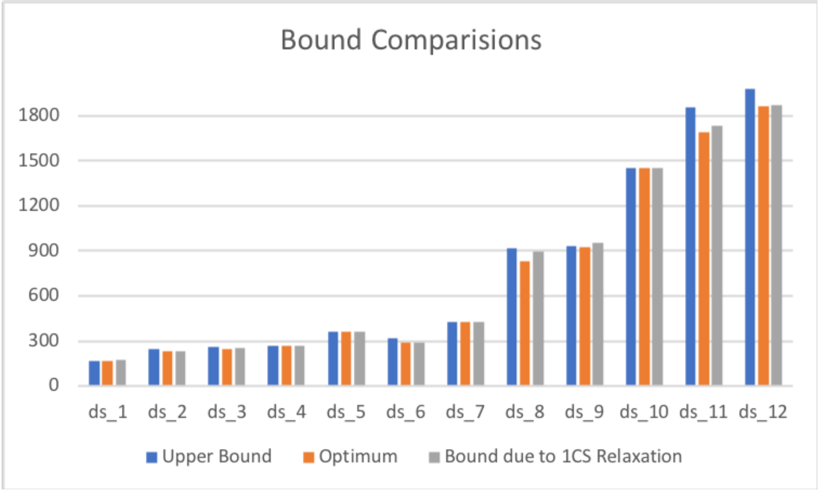


Figure: Comparision of Bounds obtained

Table of Contents

Conclusion and Future Research

Conclusion

- The empirical results reported from all the experiments based on the relaxation showed the possible profitability and benefit of the proposed approach.
- The bound obtained by LP relaxation is realtively weaker. However, we can strengthen the bound by using other solution approaches.

Directions for Future reserach

- The possible directions can be on strengthening the bound and fastening the bound computation which follows the research done in this thesis.
- Branch and Price[6] (B&P): Can be used to obtain optimal integer solutions which uses bounds from LP relaxations solved by column generation at branching nodes.
- Dual Feasible Functions[7] (DFF): Provides dual feasible solutions of models whose corresponding bounds are often very close to those provided by column generation but with very less computational burden.

[6] DAVID FRIBERG. An implementation of the branch-and-price algorithm applied to opportunistic maintenance planning.

[7] Cláudio Alves, François Clautiaux, José Carvalho, and Jürgen Rietz. DualFeasible Functions for Integer Programming and Combinatorial Optimization: Basics, Extensions and Applications. 02 2016.

Thank You!