QUESTION

(a) Use a branch and bound algorithm to solve the following (zero-one) knapsack problem. In your algorithm, always choose a node of the search tree with the largest upper bound to be explored next.

$$\begin{array}{ll} \text{Maximize} & z = 18x_1 + 17x_2 + 11x_3 + 14x_4 + 6x_5 + 4x_6 + 5x_7 \\ \text{subject to} & 8x_1 + 9x_2 + 6x_3 + 9x_4 + 4x_5 + 3x_6 + 5x_7 \leq 20 \\ & x_i = 0 \text{ or } 1 \text{ for } i = 1, \ldots, 7. \end{array}$$

Assume that the three following additional constraints are imposed:

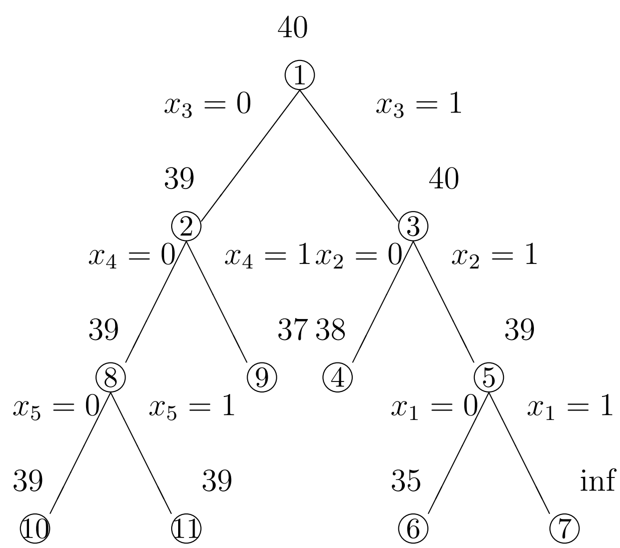
$$\begin{aligned} x_1 &\leq x_3 \\ x_2 &\leq x_4 \\ x_3 &+ x_4 + x_6 + x_7 \leq 1 \end{aligned}$$

By suitably adapting the algorithm, obtain an optimal solution to the problem with these additional constraints.

(b) Explain why the knapsack problem is useful in delayed column generation. You may explain your answer by reference to the trim loss problem: rolls of paper are cut into smaller rolls to satisfy customer demand, and the objective is to minimize the wasted paper.

ANSWER

(a) Upper bounds found by an efficient algorithm which solves linear programming relaxation.



Node 1	$UB = 18 + 17 + \lfloor \frac{11}{2} \rfloor = 40$
	$LB = 35$
Node 2	$UB = 18 + 17 + \lfloor \frac{14}{3} \rfloor = 39$
	$LB = 35$
Node 3	$UB = 18 + \lfloor \frac{2}{3} 17 \rfloor + 11 = 40$
	$LB = 29$
Node 4	$UB = 18 + \lfloor \frac{2}{3} 14 \rfloor + 11 = 38$
	$LB = 29$
Node 5	$UB = \lfloor \frac{5}{7} 18 \rfloor + 17 + 11 = 39$
	$LB = 28$
Node 6	$UB = 17 + 11 + \lfloor \frac{5}{9} 14 \rfloor = 35$
	$LB = 28$
Node 7	infeasible
Node 8	$UB = 18 + 17 + \lfloor \frac{3}{4} 6 \rfloor = 39$
	$LB = 35$
Node 9	$UB = 18 + 17 + 4 = 39$
	$LB = 32$
Node 10	$UB = 18 + 17 + 4 = 39$
	$LB = 39$
Node 11	$UB = 18 + \lfloor \frac{8}{9} 17 \rfloor + 6 = 39$
	$LB = 24$

Optimal solution at node 10,

$x_1 = x_2 = x_6 = 1$ $x_3 = x_4 = x_5 = x_7 = 0$ $z = 39$

With the additional constraints, the upper bounds remain valid. However, it may be possible to fix variables at some nodes of tjhe tree,

At nodes 4 (and 3), $x_4 = 0,\ x_2 = 0,\ x_6 = 0,\ x_7 = 0.$

Thus $UB = 18 + 11 + 6 = 35,\ LB = 35$

At node 8, $x_1 = x_2 = 0$ $UB = 6 + 4 + 5 = 15$

At node 9, $x_1 = 0$ $UB = 17 + \lfloor \frac{1}{2} 6 \rfloor + 14 = 34$

Optimal solution at node 4 $x_1 = x_3 = x_5 = 1$ $x_2 = x_4 = x_6 = x_7 = 0$ $z = 35$

(b) In the trim loss problem, every column represents a cutting combination (where the rows are the number of cuts for the different possible widths). Initially a small subset of columns is found and the linear programming problem is solved. Let y_i be the dual variable for row i. If n_i is the number of cuts for width i in a particular pattern and c is the cost, then the reduced cost is

$$c - n_1 y_1 - n_2 y_2 \ldots$$

The most negative reduced cost is given for n_i which minimize

$$c - n_1 y_1 - n_2 y_2 \ldots$$

or equivalently maximize

$$z = n_1 y_1 + n_2 y_2 \ldots \tag{1}$$

There is a constraint on the number of widths that can be cut: if w_i is the width corresponding to row i and w is the width of the original roll then

$$w_1 n_1 + w_2 n_2 + \ldots \leq w \tag{2}$$

and

$$n_1, n_2 \ldots \text{are non-negative integers} \tag{3}$$

Clearly, (1), (2), (3) defines a knapsack problem. If $z \leq c$, then there are no negative reduced cost for unconsidered cutting combinations, so the linear programming solution is optimal. Otherwise, the solution of the knapsack problem generates a new cutting pattern that is added as an extra column to the linear programming problem.