Integer Linear Programming Algorithms

Sample Solved Exercises

Departamento de Engenharia Informática (DEI)
Faculdade de Engenharia da Universidade do Porto (FEUP)

Spring 2024

Exercise 1

Consider the integer linear programming problem below with an objective function given by $\max\ 2x_1 + 2x_2$ and subject to:

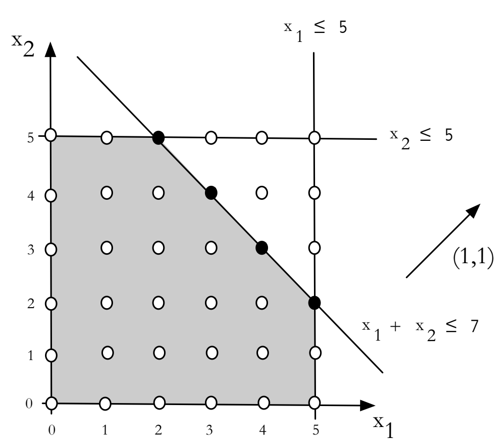
$$\begin{aligned} x_1 + x_2 &\leq 7 \\ x_1 &\leq 5 \\ x_2 &\leq 5 \end{aligned}$$

and $x_1, x_2 \geq 0$

For this ILP problem determine its feasibility, and if so, the set of optimal points and the corresponding objective function value.

Solution:

For this linear program the convex of feasible points is as shown below. Clearly, the problem is feasible and bounded. However, there are more than one optimal point, namely (2,5), (3,4), (4,3) and (5,2), all of them with a value of 14 as the maximal objective function value.



Exercises 2.
The owner of a machine shop is planning to expand by purchasing some new machines—presses and lathes. The owner has estimated that each press purchased will increase profit by $100 per day and each lathe will increase profit by $150 daily. The number of machines the owner can purchase is limited by the cost of the machines and the available floor space in the shop. The machine purchase prices and space requirements are as follows.

Machine	Required Floor Space (ft²)	Purchase Price
Press	15	$8,000
Lathe	30	4,000

The owner has a budget of $40,000 for purchasing machines and 200 square feet of available floor space. The owner wants to know how many of each type of machine to purchase to maximize the daily increase in profit.

The linear programming model for an integer programming problem is formulated in exactly the same way as the linear programming examples in chapters 2 and 4 of the text. The only difference is that in this problem, the decision variables are restricted to integer values because the owner cannot purchase a fraction, or portion, of a machine. The linear programming model follows, where x_1 is the number of presses and x_2 is the number of lathes.

$$\text{maximize } Z = 100x_1 + 150x_2$$

$$\begin{aligned} 8{,}000\,x_1 + 4{,}000\,x_2 &\leq 40{,}000 \\ 15\,x_1 + 30x_2 &\leq 200 \end{aligned}$$

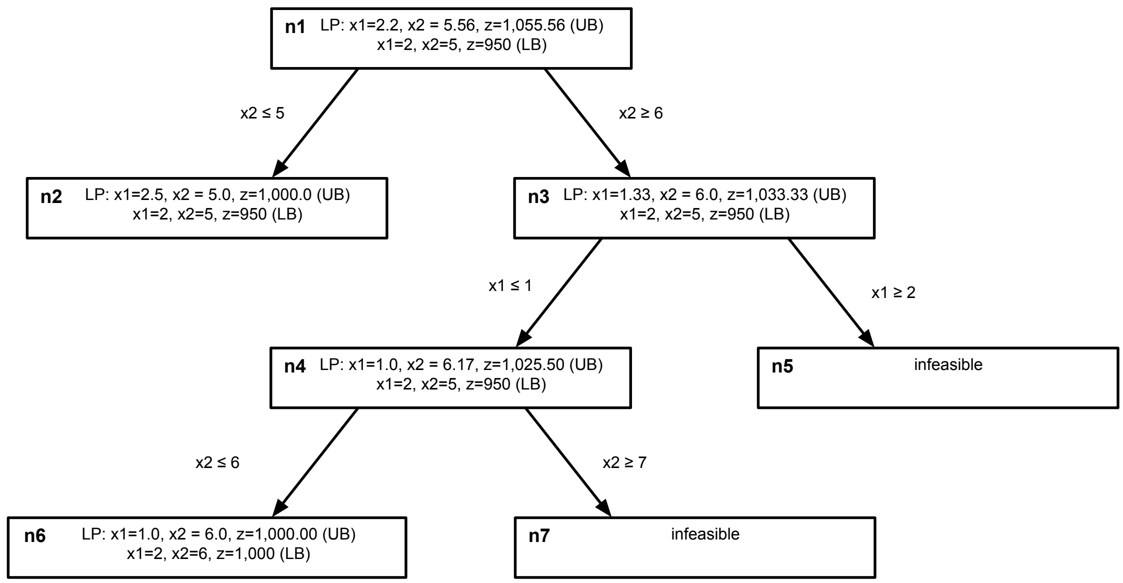
$$\text{and } x_1, x_2 \geq 0$$

The decision variables in this model are restricted to whole machines. The fact that both decision variables, x_1 and x_2, can assume any integer value greater than or equal to zero is what gives this model its designation as a total integer model.

Solution:
The state space tree diagram below depicts the state that are explored indicating for each state the current upper-bound (UB) and lower-bound (LB). For each state, the lower-bound is obtained by rounding down the value of the non-integer solution. Also, the strategy for choosing which variable to "branch-on" is to select the variable with the highest fractional part. In addition, the strategy also call for branching out of the node with the larger upper-bound as it offer the promise of larger objective function value. The order of exploration is the order of the label of the nodes.

This version of the branch and bound diagram indicates that the optimal integer solution, x1 = 1, x2 = 6, has been reached at node 6. The value of 1,000 at node 6 is the maximum, or upper bound, integer value that can be obtained. It is also the recomputed lower bound because it is the maximum integer solution achieved to this point. Thus, it is not possible to achieve any higher value by further branching from node 6. A comparison of the node 6 solution witht hose at nodes 2, 5, and 7 shows that a better solution is not possible. The upper bound at node 2 is 1,000, which is the same as that obtained at node 6; thus, node 2 can result in no improvement. The solutions at nodes 5 and 7 are infeasible (and thus further branching will result in only infeasible solutions). By the process of elimination, the integer solution at node 6 is optimal.

In general, the optimal integer solution is reached when a feasible integer solution is generated at a node and the upper bound at that node is greater than or equal to the upper bound at any other ending node (i.e., a node at the end of a branch). In the context of the original example, this solution indicates that if the machine shop owner purchases one press and six lathes, a daily increase in profit of $1,000 will result.

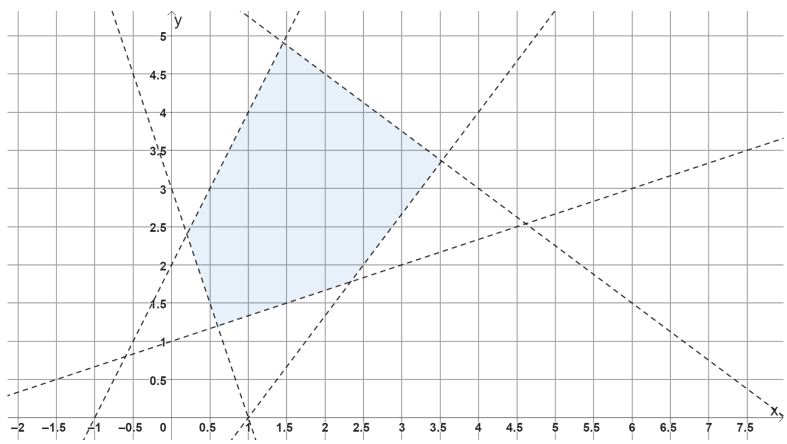


Exercises 3.

The figure below depicts the graphical representation of an Integer Programming (ILP) problem with two decision variables, x and y. A set of constraints in the problem limited the feasible region to the polygon highlighted in blue. The objective function, to be minimized, is z = x - 2y.

Using the Branch-and-Bound algorithmic approach, find the optimal solution. Explain the main steps of the algorithm by drawing a search tree and represent the optimal solution graphically in the plot.

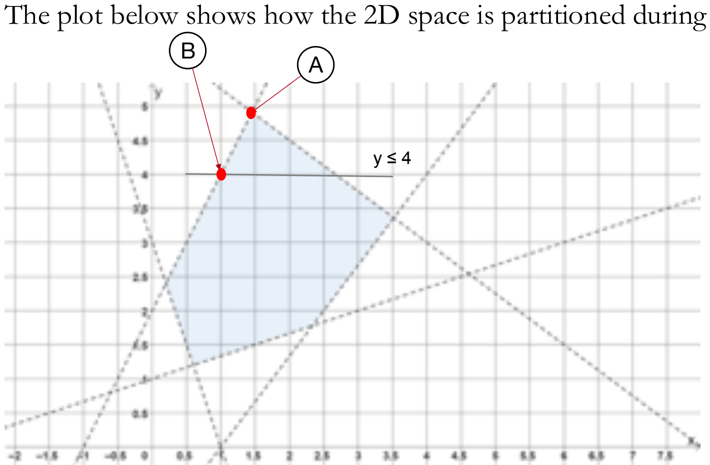
The search for the optimal solution should be conducted using Depth-First Search (DFS).

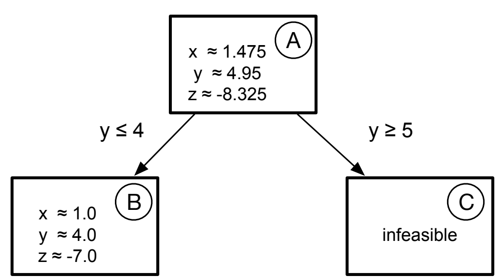


Solution:

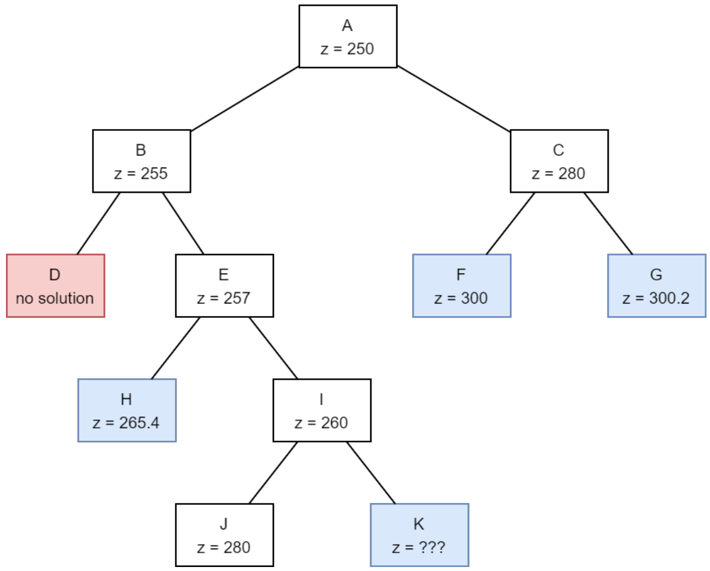
The image below presents the search tree generated via backtracking during the Branch-and-Bound algorithm. In this case, the tree is very short, as during the first partition of the space, we get an integer solution right away. Note that the LP solution, A = (1.475, 4.95) with objective value z = -8.325 which is a lower-bound on the optimal minimal value. After constraining the space by $y \leq 4$ we get B = (1.0, 4.0) which yields an objective value of z = -7.0. As this is an integer value, it is an upper-bound on the solution. We can still explore a possible solution for $y \geq 5$. However, this results in an infeasible solution, and hence solution at B is optimal.

The plot below shows how the 2D space is partitioned during the search.





Exercises 4.
A computer running the Branch-and-Bound algorithm for an Integer Linear Programming problem produced the search tree below. This problem has an objective function known as z.



The nodes of the search tree were explored in alphabetical order. The nodes in blue correspond to all-integer solutions, while the node in red represents an inadmissible solution to the relaxed Linear Programming problem.

a) Determine and justify whether this is a maximization or minimization problem.
b) Assuming node K contains the optimal solution, determine this node's lower and upper bounds for z.
c) Which algorithm was used to explore the solution space (Depth-First Search or Breadth-First Search)? Justify your answer. Is this strategy the best suited one for this problem?

Solution:

a) This is a minimization problem since z increases as more constraints are added, i.e. as the node depth increases in the search tree.
b) When node K is reached, the best solution found so far is on node H with z = 265.4. Therefore, the solution in node K must have a z that is lower or equal than 265.4 (265.4 is thus the upper bound)

The father of node K is I, with z = 260. The relaxed LP problems in K and I differ by K having one additional constraint, thus z's value for K cannot be better (i.e. smaller) than the one in I, since the space of admissible solutions for K is a subset of the one for I. Therefore, the solution in node K must have a z that is higher or equal than 260 (260 is thus the lower bound).

To conclude, in node K, z ∈ [260, 265.4].

c) The search was conducted using Breadth-First Search (BFS). This can be seen for instance in the root, where each child is explored before reaching the root's grandchildren.

Using Depth-First Search (DFS) is be more suitable for this problem as avoids having to solve the relaxed LP problems of nodes F and G. During the DFS, the solution of node H is found before reaching C, so C's children would be pruned since node C's z value is already higher than the one for H, which implies that F and G would have sub-optimal (i.e. higher) z values.