Department of Mathematics and Computer Science
University of Southern Denmark, Odense

October 20, 2025
Marco Chiarandini

DM545/DM871 – Linear and integer programming

Sheet 6, Autumn 2025

Exercises with the symbol $^+$ are to be done at home before the class. Exercises with the symbol * will be tackled in class and should be at least read at home. The remaining exercises are left for self training after the exercise class.

Solution:

Included.

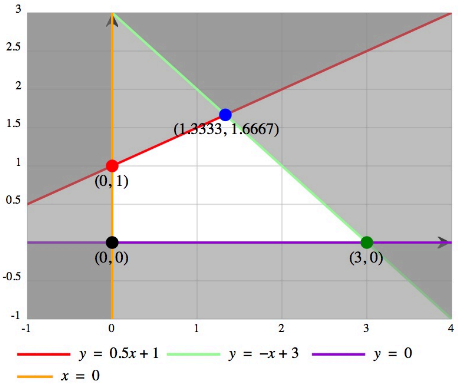
Exercise 1$^+$

Try solving the following IP problem with the Gomory's fractional cutting plane algorithm, indicating the cut inequalities in the space of the original variables

$$\begin{aligned} \max\ & x_1 + 2x_2 \\ & x_1 - 2x_2 \geq -2 \\ & x_1 + x_2 \leq 3 \\ & x_1, x_2 \geq 0 \text{ and integer} \end{aligned}$$

Solution:

We represent the situation graphically using the LP Grapher tool linked from the web page (any other graphing tool like Grapher under Macosx would do a similar job)



```
|-------+-------+-------+-------+-------+-------+
|    x1 |    x2 |    x3 |    x4 |    -z |     b |
|-------+-------+-------+-------+-------+-------+
|    -1 |     2 |     1 |     0 |     0 |     2 |
|     1 |     1 |     0 |     1 |     0 |     3 |
|-------+-------+-------+-------+-------+-------+
|     1 |     2 |     0 |     0 |     1 |     0 |
|-------+-------+-------+-------+-------+-------+
pivot column: 2
pivot row: 1
 pivot: 2

|-------+-------+-------+-------+-------+-------+
|    x1 |    x2 |    x3 |    x4 |    -z |     b |
|-------+-------+-------+-------+-------+-------+
|  -1/2 |     1 |   1/2 |     0 |     0 |     1 |
|   3/2 |     0 |  -1/2 |     1 |     0 |     2 |
|-------+-------+-------+-------+-------+-------+
|     2 |     0 |    -1 |     0 |     1 |    -2 |
|-------+-------+-------+-------+-------+-------+
pivot column: 1
pivot row: 2
 pivot: 3/2

|-------+-------+-------+-------+-------+-------+
|    x1 |    x2 |    x3 |    x4 |    -z |     b |
|-------+-------+-------+-------+-------+-------+
|     0 |     1 |   1/3 |   1/3 |     0 |   5/3 |
|     1 |     0 |  -1/3 |   2/3 |     0 |   4/3 |
|-------+-------+-------+-------+-------+-------+
|     0 |     0 |  -1/3 |  -4/3 |     1 | -14/3 |
|-------+-------+-------+-------+-------+-------+
```

We choose the first row. The cut is:

$$1/3x_3 + 1/3x_4 \geq 2/3$$

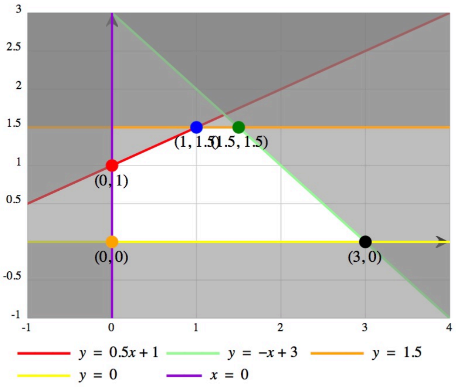
To express the cut in terms of the original variables we need to express the variables x_3 and x_4 as a function of the other two that are in basis. We consier the equational standard form of the problem, which corresponds to the initial tableau. From there we see that:

$$x_3 = 2 + x_1 - 2x_2$$
$$x_4 = 3 - x_1 - x_2$$

and substituting in the cut we get:

$$x_2 \leq 1$$

The cut separates the LP solution as evident from the picture below:



We insert the cut in the simplex. We can insert either the cut in the non basic variables or the cut in the original variables. If we choose the cut in the non basic variables the tableau will be already in canonical form but infeasible:

```
|-------+-------+-------+-------+-------+-------+-------+
|    x1 |    x2 |    x3 |    x4 |    x5 |    -z |     b |
|-------+-------+-------+-------+-------+-------+-------+
|     0 |     0 |  -1/3 |  -1/3 |     1 |     0 |  -2/3 |
|     0 |     1 |   1/3 |   1/3 |     0 |     0 |   5/3 |
|     1 |     0 |  -1/3 |   2/3 |     0 |     0 |   4/3 |
|-------+-------+-------+-------+-------+-------+-------+
|     0 |     0 |  -1/3 |  -4/3 |     0 |     1 | -14/3 |
|-------+-------+-------+-------+-------+-------+-------+
```

If we insert the cut in the form $x_2 \leq 1$ we would need to put the tableau in canonical form before proceeding.
The tableau is infeasible and we need to continue with the dual simplex. The pivot row is the first one and the pivot column is $\text{argmin}_j\{|c_j|/|a_{ij}| \mid a_{ij} < 0\} = 3$. After the pivoting operations we obtain:

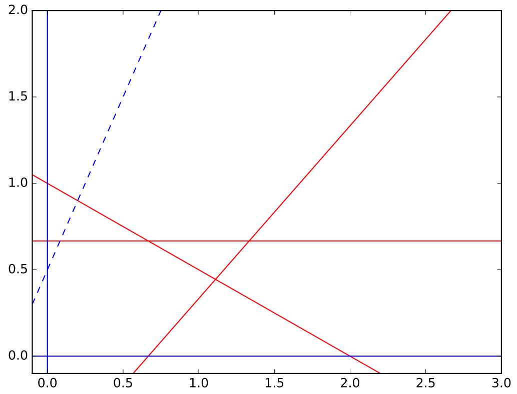
```
|----+----+----+----+----+----+----|
| x1 | x2 | x3 | x4 | x5 | -z |  b |
|----+----+----+----+----+----+----|
|  0 |  0 |  1 |  1 | -3 |  0 |  2 |
|  0 |  1 |  0 |  0 |  1 |  0 |  1 |
|  1 |  0 |  0 |  1 | -1 |  0 |  2 |
|----+----+----+----+----+----+----|
|  0 |  0 |  0 | -1 | -1 |  1 | -4 |
|----+----+----+----+----+----+----|
```

The optimal solution is now integer $(x_1 = 2, x_2 = 1)$ and therefore we solved the original integer programming problem.

Exercise 2* — Gomory's Cutting Plane

Consider the following integer linear programming problem

$$\begin{array}{lrcrl}
\max & z = 4x_0 & - & 2x_1 & \\
\text{s.t.} & x_0 & + & 2x_1 & \leq 2 \\
 & & & 3x_1 & \leq 2 \\
 & 3x_0 & - & 3x_1 & \leq 2 \\
 & & & x_0, x_1 & \geq 0 \text{ and integer}
\end{array}$$



In the solution of the linear relaxation of the problem the variables x_0, x_1 and the slack variable associated to the second constraint are in basis.

The data in Python format:

```
#%%
from fractions import Fraction as f
import numpy as np
np.set_printoptions(precision=3,suppress=True)

c=np.array([4, -2])
A = np.array([[ 1, 2],
              [ 0, 3],
              [ 3, -3]])

b=np.array([2, 2, 2])
```

Subtask 2.1

Calculate the optimal tableau using the revised simplex method.

Solution:

Let x_2, x_3, x_4 be the slack variables in the equational standard form. Recalling the theory of the revised simplex, we can calculate the final table as follows:

$$\left[\begin{array}{c:c:c:c} I & \bar{A}_N = A_B^{-1}A_N & 0 & \bar{b} \\ \hdashline \bar{\mathbf{c}}_B = \mathbf{0} & \bar{\mathbf{c}}_N = \mathbf{c}_N - \mathbf{c}_B A_B^{-1}\mathbf{b}(\leq 0) & 1 & -\bar{d} \end{array}\right]$$

Hence we need to determine $\bar{b}$ and $\bar{A}_N = A_B^{-1}A_N$. In Python:

```
AI = np.concatenate([A,np.identity(3)],axis=1)
ci=np.concatenate([c,[0,0,0]])

basis = np.array([0,1,3])
nonbasis = np.array([2,4])
# basis = np.array([1,2,3])
# nonbasis = np.array([0,4])
B = AI[:,basis]
N = AI[:,nonbasis]

B_i = np.linalg.inv(B)

x_B = np.dot(B_i,b)
print(x_B)

#%%
y = np.dot(ci[basis],B_i)
red_costs = ci[nonbasis]-np.dot(y,N)
print( red_costs)

# %%
print(np.dot(ci[basis],x_B))
# %%
print(np.dot(B_i,N))
```

```
[ 1.111  0.444  0.667]
[-0.667 -1.111]
3.55555555556
[[ 0.333  0.222]
 [ 0.333 -0.111]
 [-1.     0.333]]
```

Hence the tableau looks as follows:

```
|----+----+--------+----+--------+----+--------|
| x0 | x1 |     x2 | x3 |     x4 | -z |      b |
|----+----+--------+----+--------+----+--------|
|  1 |  0 |  0.333 |  0 |  0.222 |  0 |  1.111 |
|  0 |  1 |  0.333 |  0 | -0.111 |  0 |  0.444 |
|  0 |  0 |     -1 |  1 |  0.333 |  0 |  0.667 |
|----+----+--------+----+--------+----+--------|
|  0 |  0 | -0.667 |  0 | -1.111 |  1 | -3.555 |
|----+----+--------+----+--------+----+--------|
```

Similarly, we could have used one of the available tools in fractional mode and obtained:

```
|--------+--------+--------+--------+--------+--------+--------+
|     x0 |     x1 |     x2 |     x3 |     x4 |     -z |      b |
|--------+--------+--------+--------+--------+--------+--------+
|      1 |      2 |      1 |      0 |      0 |      0 |      2 |
|      0 |      3 |      0 |      1 |      0 |      0 |      2 |
|      3 |     -3 |      0 |      0 |      1 |      0 |      2 |
|--------+--------+--------+--------+--------+--------+--------+
|      4 |     -2 |      0 |      0 |      0 |      1 |      0 |
|--------+--------+--------+--------+--------+--------+--------+
PRIMAL SIMPLEX
Confirm pivot column: 00

pivot column: 1
pivot row: 3
 pivot: 3
3
|--------+--------+--------+--------+--------+--------+--------+
|     x0 |     x1 |     x2 |     x3 |     x4 |     -z |      b |
|--------+--------+--------+--------+--------+--------+--------+
|      0 |      3 |      1 |      0 |   -1/3 |      0 |    4/3 |
|      0 |      3 |      0 |      1 |      0 |      0 |      2 |
|      1 |     -1 |      0 |      0 |    1/3 |      0 |    2/3 |
|--------+--------+--------+--------+--------+--------+--------+
|      0 |      2 |      0 |      0 |   -4/3 |      1 |   -8/3 |
|--------+--------+--------+--------+--------+--------+--------+
Confirm pivot column: 11

pivot column: 2
pivot row: 1
 pivot: 3
3
|--------+--------+--------+--------+--------+--------+--------+
|     x0 |     x1 |     x2 |     x3 |     x4 |     -z |      b |
|--------+--------+--------+--------+--------+--------+--------+
|      0 |      1 |    1/3 |      0 |   -1/9 |      0 |    4/9 |
|      0 |      0 |     -1 |      1 |    1/3 |      0 |    2/3 |
|      1 |      0 |    1/3 |      0 |    2/9 |      0 |   10/9 |
|--------+--------+--------+--------+--------+--------+--------+
|      0 |      0 |   -2/3 |      0 |  -10/9 |      1 |  -32/9 |
|--------+--------+--------+--------+--------+--------+--------+
```

Subtask 2.2

Find a Chvatal Gomory's cutting plane

Solution:

We consider the Gomory cut relative to the variable $x_0 = 1.111$.

$$\sum_{j \in N} (\bar{a}_{uj} - \lfloor \bar{a}_{uj} \rfloor) x_j \geq \bar{b}_u - \lfloor \bar{b}_u \rfloor$$

and substituting the numbers:

$$1/3x_2 + 2/9x_4 \geq 1/9$$

or

$$0.333x_2 + 0.222x_4 \geq 0.111$$

Subtask 2.3

Show that with the cut found the optimal solution of the linear relaxation becomes infeasible.

Solution:

The optimal solution has $x_2 = 0$ and $x_4 = 0$, hence inserting the cut in the problem the optimal solution becomes infeasible.

Exercise 3* — Branch and Bound

This exercise is taken from the exam of 2012.
The Danish Research Council has to decide which research projects to finance. The total budget for the projects is 20 million Dkk. The table below shows the evaluation from 0 (worst) to 2 (best) that the projects received by the external reviewers and the amount of money required.

	1	2	3	4	5
Evaluation score	1	1.8	1.4	0.6	1.4
Investment (in million of DKK)	6	12	10	4	8

Projects 2 and 3 have the same coordinator and the Council decided to grant only one of the two.
The Council wants to select the combination of projects that will maximize the total relevance of the projects, that is, the sum of the evaluation score while remaining within the budget.

Subtask 3.4

Formulate the problem of deciding on which project the Council has to invest as an integer linear programming problem P.

Solution:
In .lp format:

```
\* Problem: lp3 *\

Maximize
 tot: + x(1) + 1.8 x(2) + 1.4 x(3) + 0.6 x(4) + 1.4 x(5)

Subject To
 budget: + 6 x(1) + 12 x(2) + 10 x(3) + 4 x(4) + 8 x(5) <= 20
 a: + x(2) + x(3) <= 1

Bounds
 0 <= x(1) <= 1
 0 <= x(2) <= 1
 0 <= x(3) <= 1
 0 <= x(4) <= 1
 0 <= x(5) <= 1

End
```

Subtask 3.5

We want the IP instance solved using the branch-and-bound algorithm. What is the optimal solution x^* to the LP relaxation P'? (Hint: use the tool: http://www.zweigmedia.com/simplex/simplex.php to solve the LP problems.]

Solution:
The following gurobipy model gives solution $\mathbf{x} = [1, 0.166, 0, 1, 1]$ and $z = 3.3$.

```
import gurobipy as gp

m = gp.Model("knapsack")

s=[1,1.8,1.4,0.6,1.4]
c=[6,12,10,4,8]


x = {i:m.addVar(lb=0.0, ub=1.0, vtype=gp.GRB.CONTINUOUS, name="x%d" % i) for i in range
    (1,6)}
```

```
m.setObjective(gp.quicksum(s[i-1]*x[i] for i in range(1,6)), gp.GRB.MAXIMIZE)

m.addConstr(gp.quicksum(c[i-1]*x[i] for i in range(1,6))<=20, "c1")
m.addConstr(x[2]+x[3] <= 1, "c2")

m.write("knapsack.lp")
m.write("knapsack.mps")

m.optimize()

print({v.varname: v.x for v in m.getVars() })
```

```
Optimize a model with 2 rows, 5 columns and 7 nonzeros
Coefficient statistics:
  Matrix range     [1e+00, 1e+01]
  Objective range [6e-01, 2e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 2e+01]
Presolve time: 0.00s
Presolved: 2 rows, 5 columns, 7 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    6.2000000e+00   2.250000e+00   0.000000e+00      0s
       2    3.3000000e+00   0.000000e+00   0.000000e+00      0s

Solved in 2 iterations and 0.00 seconds
Optimal objective  3.300000000e+00
[1.0, 0.166, 0.0, 1.0, 1.0]
```

Subtask 3.6

The rounding heuristic applied to the solution x^* gives a feasible solution x'. Which one? With the knowledge collected until this stage which of the three following statements is correct:

1. x' is certainly optimal
2. x' is certainly not optimal
3. x' might be optimal

(Remember to justify your answer.)

Solution:

The rounding heuristic updates x^* setting $x_2 = 0$ or $x_2 = 1$. The latter gives an infeasible solution while the former gives $[1, 0, 0, 1, 1]$ with value 3. We cannot say at this stage if x' is optimal because the optimality gap 3.3–3 is not closed. Hence (iii) is correct.

Subtask 3.7

The two subproblems generated by the branch-and-bound algorithm after finding x^* correspond to choosing or not choosing a particular project. Which one?

Solution:

The solution is $[1, 0.166, 0, 1, 1]$ and the only fractional variable is x_2 hence we branch on it.

Subtask 3.8

Suppose the branch-and-bound algorithm considers first the subproblem corresponding to not choosing this project. Let's call this subproblem and its corresponding node in the search tree SP1. What is the optimal solution to its LP relaxation?

Solution:

Adding the constraint `x2<=0` and solving again we obtain:

$$x = [1, 0, 0.2, 1, 1]$$

and $z = 3.28$.

Subtask 3.9

Next, the branch-and-bound algorithm considers the subproblem corresponding to choosing the project, i.e., subproblem SP2. Find the optimal solution to its LP relaxation. Which are the active nodes (i.e., open subproblems) at this point?

Solution:

Adding the constraint `x2>=1` to the Python code above we obtain:

$$x = [0, 1, 0, 0, 1]$$

and $z = 3.2$. This is an integer solution and hence a lower bound.

Node SP2 is not active since an integer solution prunes the subtree. The other node SP1 has however still potential to find a better solution since its upper bound is $3.28 > 3.2$, hence the list of active nodes contains SP1.

Subtask 3.10

How does the branch and bound end?

Solution:

We need to examine the active nodes. Hence we branch once more with $x_3 \leq 0$ (subproblem SP3) and $x_3 \geq 1$ (subproblem SP4). The LP relaxation of SP3 gives an integer solution $[1, 0, 0, 1, 1]$ of value 3 and SP4 gives $[0.33, 0, 1, 0, 1]$ of value 3.13. Hence the upper bound from subtree SP1 is 3.13 which is smaller than the lower bound 3.2 of SP2 and we can prune SP4 by bounding. The optimal solution is the one on node SP2.

Exercise 4 — Branch and bound

Consider the following ILP:

$$\begin{array}{llrrrrrrrrrrrl}
\max & z = & 5x_1 & + & 5x_2 & + & 8x_3 & - & 2x_4 & - & 4x_5 & & \\
& \text{s.t.} & -3x_1 & + & 6x_2 & - & 7x_3 & + & 9x_4 & + & 9x_5 & \geq & 10 \\
& & x_1 & + & 2x_2 & & & - & x_4 & - & 3x_5 & \leq & 0 \\
& & \multicolumn{11}{l}{x_1, x_2, x_3, x_4, x_5 \in \{0, 1\}}
\end{array}$$

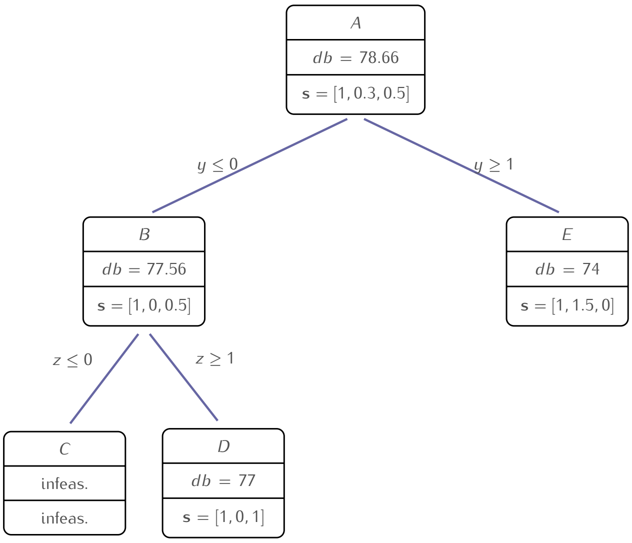
Solve the problem by branch and bound:

- Use objective-value bounding for pruning subproblems.
- At each node use linear programming to find dual bounds.
- Use the *most fractional variable* rule for branching.
- Follow a depth first search strategy and expand first the *greater-or-equal* branch.

Answer guidelines:

- Make sure that you indicate which is the final solution and its objective function value.
- Use this tool to solve the linear relaxations at each node:
  http://www.zweigmedia.com/simplex/simplex.php
- In the next pages you are given a template for the search tree. Write the search tree first on the paper version of the exam that you received and then digitalize your answer in one of the following ways:
  - scan the tree that you have handwritten (make sure you write all the information needed — see example in the next pages)
  - annotate the tree template provided in the next pages, make a screenshot and include it in your document.
  - use the text format explained in the next page.

Other reporting examples
Drawing: The following is an example of drawing for describing the search tree. The example is not taken from the problem object of this task.



Text format: The search tree above can be described in text format as shown below.

```
---
- name: A
  parent: ’null’
  constraint_added: ’’
  dual_bound: 78.66
  solution: [1, 0.3, 0.5]
  children:
  - name: B
    parent: A
    constraint_added: y<=0
    dual_bound: 77.56
    solution: [1,0,0.5]
    children:
    - name: C
      parent: B
      constraint_added: z<=0
      dual_bound: infeasible
      solution: infeasible
      children: pruned
    - name: D
      parent: B
      constraint_added: z>=1
```

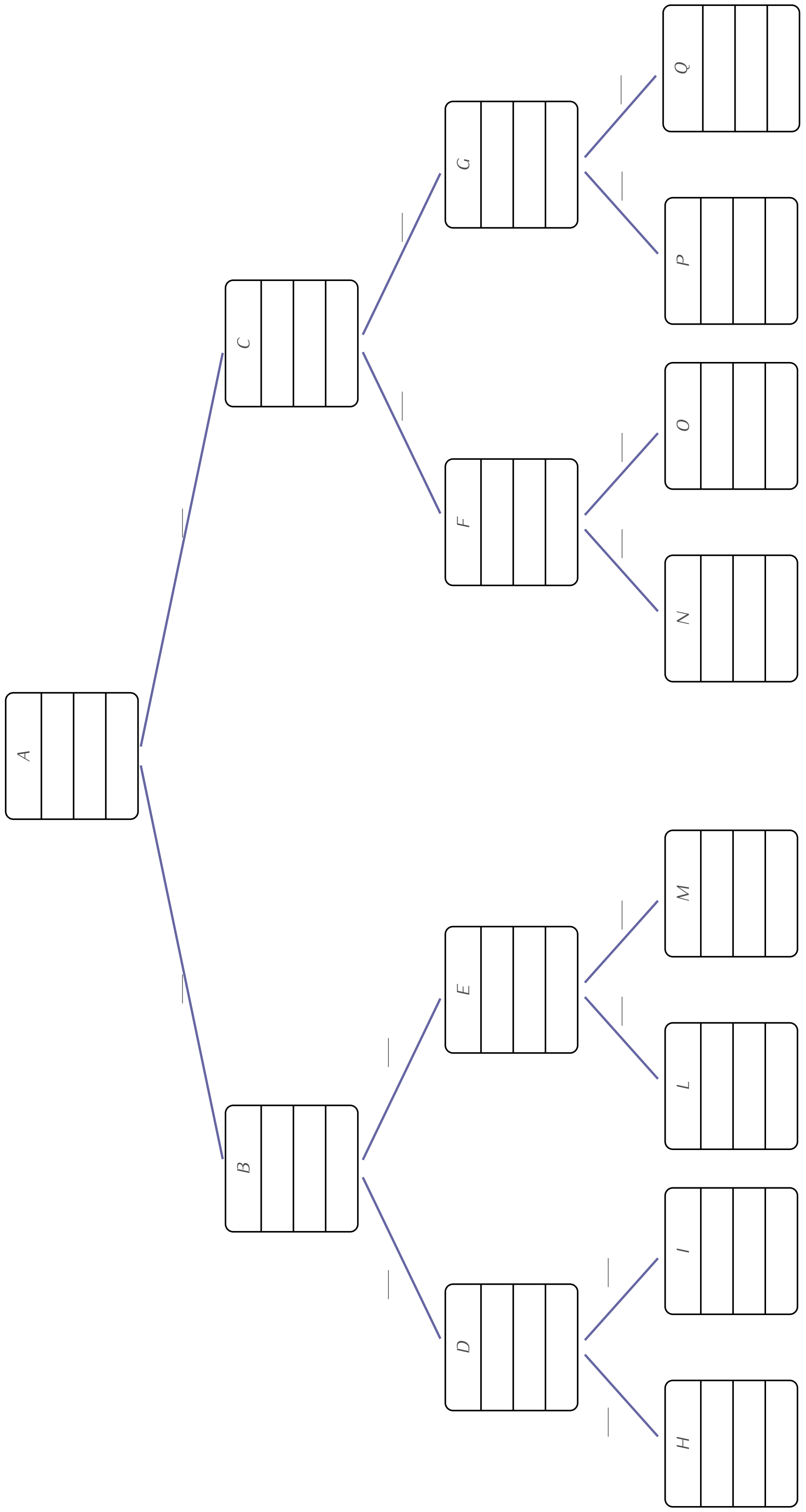


Figure 1: A template for a search tree that you can annotate

```
     dual_bound: 77
     solution: [1,0,1]
     children: pruned
  - name: E
    parent: A
    constraint_added: y>=1
    dual_bound: 74
    solution: [1, 1.5, 0]
    children: pruned
```

Solution:

The model in Python is:

```
#!/usr/bin/python
from gurobipy import *

# Model
model = Model("prod")
model.setParam(GRB.param.Method, 0)
model.setParam(GRB.param.Presolve, 0)

## Add here the LP model
## See Sheet2 from Linear and Integer Programming Part for an example of the syntax


# Create decision variables
x1 = model.addVar(lb=0.0, ub=1.0, vtype=GRB.CONTINUOUS, name="x1")
x2 = model.addVar(lb=0.0, ub=1.0, vtype=GRB.CONTINUOUS, name="x2")
x3 = model.addVar(lb=0.0, ub=1.0, vtype=GRB.CONTINUOUS, name="x3")
x4 = model.addVar(lb=0.0, ub=1.0, vtype=GRB.CONTINUOUS, name="x4")
x5 = model.addVar(lb=0.0, ub=1.0, vtype=GRB.CONTINUOUS, name="x5")

model.update()


# The objective is to maximize (this is redundant now, but it will overwrite Var
    declaration
model.setObjective(5*x1 + 5*x2 + 8*x3 -2*x4-4*x5, GRB.MAXIMIZE)

# Add constraints to the model
model.addConstr(-3*x1 + 6*x2 - 7*x3 + 9*x4+9*x5, GRB.GREATER_EQUAL, 10.0, "c1")
model.addConstr(x1 + 2*x2 - x4-3*x5, GRB.LESS_EQUAL, 0.0, "c1")

#model.addConstr(x5 , GRB.LESS_EQUAL, 0.0, "c1")
##model.addConstr(x5 , GRB.GREATER_EQUAL, 1.0, "c1"
#model.addConstr(x2 , GRB.GREATER_EQUAL, 1.0, "c1")

#model.addConstr(x5 , GRB.GREATER_EQUAL, 1.0, "c1")
#model.addConstr(x4 , GRB.GREATER_EQUAL, 1.0, "c1")
#model.addConstr(x4 , GRB.LESS_EQUAL, 0.0, "c1")

# Solve
model.optimize()




if model.status == GRB.status.OPTIMAL:
    # Let us print the solution
    for v in model.getVars():
        print( v.varName, v.x)
    print("dual_bound: %g" % round(model.objVal,3))
    print("solution: "+str( [round(v.x,3) for v in model.getVars()]))
else:
    print("Optimization was stopped with status %d" % model.status)
```

The final tree is given in Figure 2.
The solution process is pruned at node F by bounding, indeed the upper bound down that sub tree is smaller than the incumbent solution found because of integrality of solutions in D. Similarly the solution process is pruned at node B by bounding. In the figure nodes D and E are expanded anyway but they should not. The search proceeded in this order: A, C, G, F, B. At the root we branched on the variable x_5 because more fractional than x_4.
The optimal solution is found in node G. It has value 12 and it is $\mathbf{x} = [1; 1; 1; 1; 1]$.
This is confirmed by Gurobi changing the definition of the variables to be from GRB.CONTINUOUS to be GRB.BINARY.

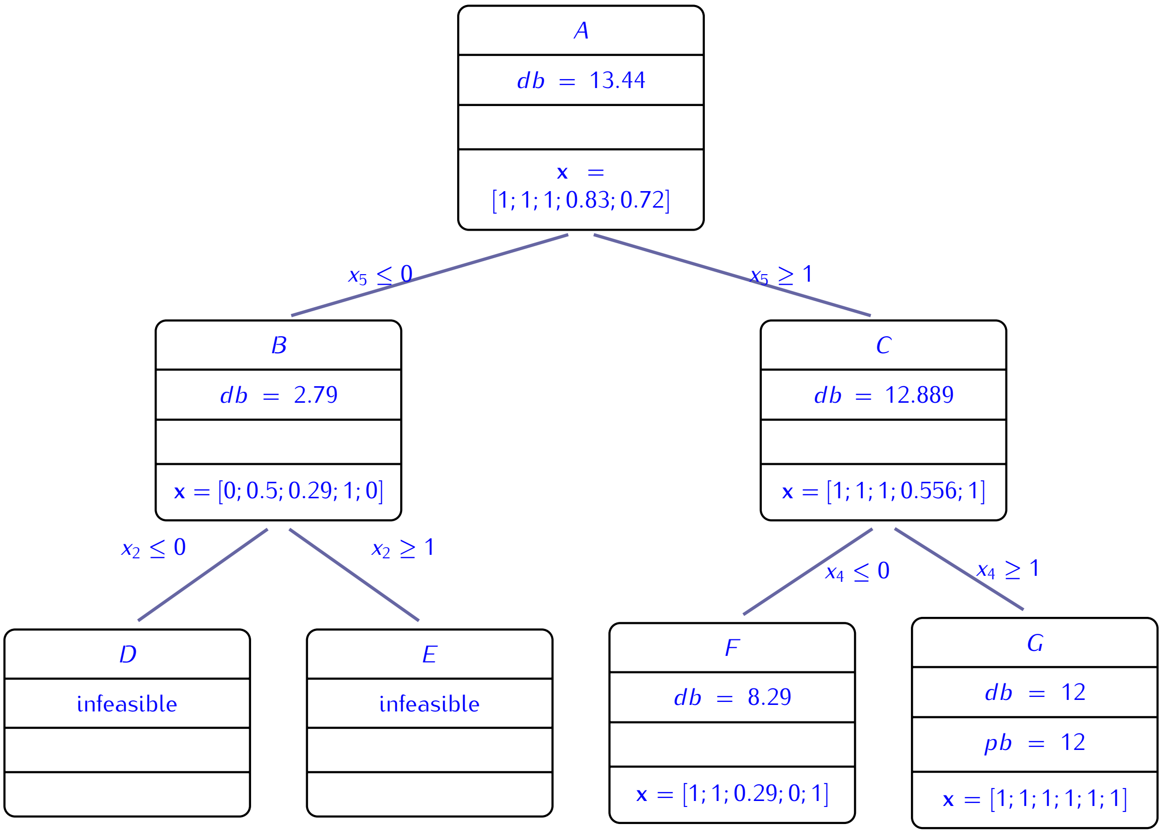

A
$db = 13.44$
$\mathbf{x} = [1; 1; 1; 0.83; 0.72]$
$x_5 \leq 0$
$x_5 \geq 1$
B
$db = 2.79$
$\mathbf{x} = [0; 0.5; 0.29; 1; 0]$
C
$db = 12.889$
$\mathbf{x} = [1; 1; 1; 0.556; 1]$
$x_2 \leq 0$
$x_2 \geq 1$
$x_4 \leq 0$
$x_4 \geq 1$
D
infeasible
E
infeasible
F
$db = 8.29$
$\mathbf{x} = [1; 1; 0.29; 0; 1]$
G
$db = 12$
$pb = 12$
$\mathbf{x} = [1; 1; 1; 1; 1]$

Figure 2: