Solution Procedures: The Branch & Bound Method

(This works for any MILP, in principle, and is theoretically exact. These problems are, however, <u>much</u> more difficult to solve efficiently than pure LP, and solution efficiency is much more sensitive to particular problem characteristics)

Consider a MILP; with binary variables (this works for the general case, but easier to describe for 0/1)

$$\min_{x,y} \; c^T \begin{Bmatrix} x \\ y \end{Bmatrix} \tag{2}$$

$$\text{s.t.} \quad A \begin{Bmatrix} x \\ y \end{Bmatrix} \geq b$$

$$x \geq 0$$

$$y = \{0, 1\}$$

And define a relaxed MILP, where the binary variables are allowed to be continuous:

$$\min_{x,y} \; c^T \begin{Bmatrix} x \\ y \end{Bmatrix} \tag{2R}$$

$$\text{s.t.} \quad A \begin{Bmatrix} x \\ y \end{Bmatrix} \geq b$$

$$x \geq 0$$

$$0 \leq y \leq 1$$

Let the solution to (2R) have the optimal objective value f_R^*, and define the solution to (2) to have the optimal objective value f^*.

Question: Is $f_R^* \leq f^*$ always true?

(i.e. is the relaxed problem a bound on the solution to the real problem).

Answer: YES! ADDING A CONSTRAINT (such as making $y = \{0,1\}$) can never improve the situation. If it were optimal for $y = \{0,1\}$ then that would also be the case for the relaxed problem.

We will use this fact in the B&B method to prune the search tree and thus limit the number of cases that need to be investigated.

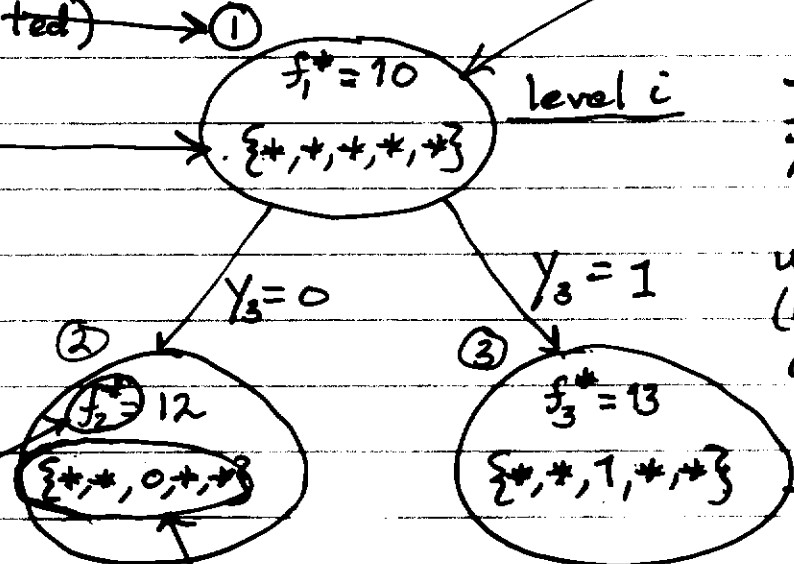
Let's explore the B&B method via an example. This example has 5 integer binary variables y_1, y_2, y_3, y_4, y_5.

FIRST DEFINE THE Search TREE

The 'Root' node

Node number (order visited)

A node in the tree (a particular relaxed problem)



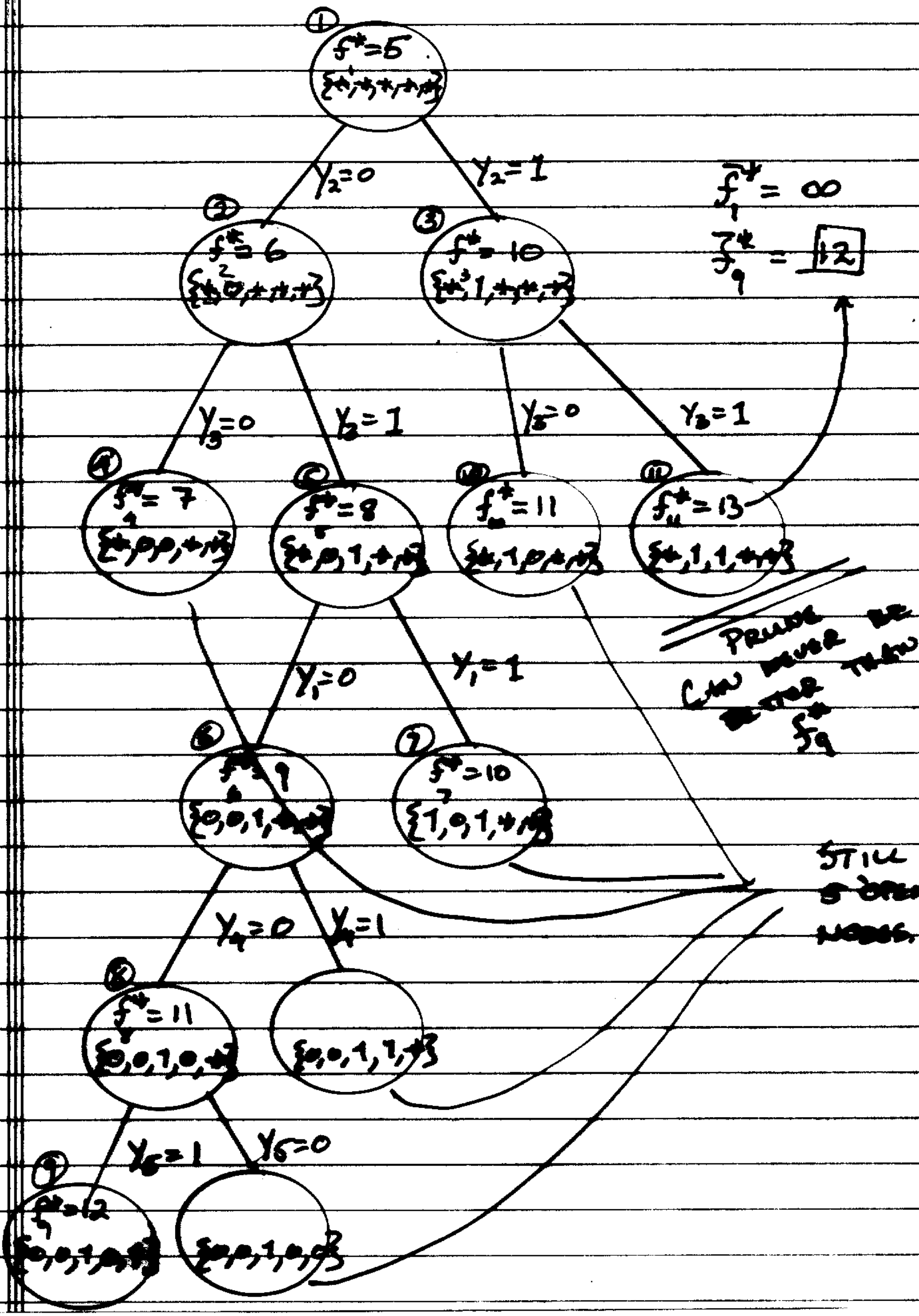
$\bar{f}_i^* = \square$

upper bound (worst case) on optimal solution to MILP (after node j)

optimal solution for relaxed problem at node

which integer variables are fixed at integer values. (others are free in interval [0, 1])

Moving 1 level on the tree means fixing one 0/1 variable at its integer values. Thus for an n-variable (binary) problem, we have n levels and 2^n possible nodes.



1
f* = 5
{*, *, *, *, *}
Y2 = 0
Y2 = 1
2
f2* = 6
{*, 0, *, *, *}
3
f3* = 10
{*, 1, *, *, *}
f1* = ∞
f9* = 12
Y3 = 0
Y3 = 1
Y3 = 0
Y3 = 1
4
f4* = 7
{*, 0, 0, *, *}
5
f5* = 8
{*, 0, 1, *, *}
10
f10* = 11
{*, 1, 0, *, *}
11
f11* = 13
{*, 1, 1, *, *}
PRUNE
CAN NEVER BE BETTER THAN
f9*
Y1 = 0
Y1 = 1
6
f6* = 9
{0, 0, 1, *, *}
7
f7* = 10
{1, 0, 1, *, *}
STILL
5 OPEN
NODES.
Y4 = 0
Y4 = 1
8
f8* = 11
{0, 0, 1, 0, *}
{0, 0, 1, 1, *}
Y5 = 1
Y5 = 0
9
f9* = 12
{0, 0, 1, 0, 1}
{0, 0, 1, 0, 0}