ORF 522: Lecture 20

Integer Programming

Robert J. Vanderbei

December 5, 2013

Slides last edited at 11:08am on Thursday 5^{th} December, 2013

Operations Research and Financial Engineering, Princeton University

http://www.princeton.edu/∼rvdb

Airline Equipment Scheduling

Given:

- A set of *flight legs* (e.g. Newark to Chicago departing 7:45am).
- A set of aircraft.

Problem: which specific aircraft should fly which flight legs?

Model:

- Generate a set of feasible *routes* (i.e., a collection of legs which taken together can be flown by one airplane).
- Assign a cost to each route (e.g. 1).
- Pick a minimum cost collection of routes that exactly covers all of the legs.

Let:

$$x_j = \begin{cases} 1 & \text{if route } j \text{ is selected,} \\ 0 & \text{otherwise} \end{cases}$$

$$a_{ij} = \begin{cases} 1 & \text{if leg } i \text{ is part of route } j, \\ 0 & \text{otherwise} \end{cases}$$

$$c_j = \text{cost of using route } j.$$

An Integer Programming Problem:

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{j=1}^{n} c_j x_j & \\ \text{subject to} & \displaystyle\sum_{j=1}^{n} a_{ij} x_j = 1 & i = 1, 2, \ldots, m, \\ & x_j \in \{0, 1\} & j = 1, 2, \ldots, n. \end{array}$$

An example of *set-partitioning problems*.

Airline Crew Scheduling

Similar to equipment scheduling except:

It's possible to put more than one crew on a flight:

- only one crew works
- any others are just being shuttled

Integer Programming Problem:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{j=1}^{n} c_j x_j \\
\text{subject to} \quad & \sum_{j=1}^{n} a_{ij} x_j \geq 1 && i = 1, 2, \ldots, m, \\
& x_j \in \{0, 1\} && j = 1, 2, \ldots, n.
\end{aligned}
$$

An example of *set-covering problems*.

Column Generation

The problem of producing a set of possible routes is called *column generation.*

It is important and nontrivial.

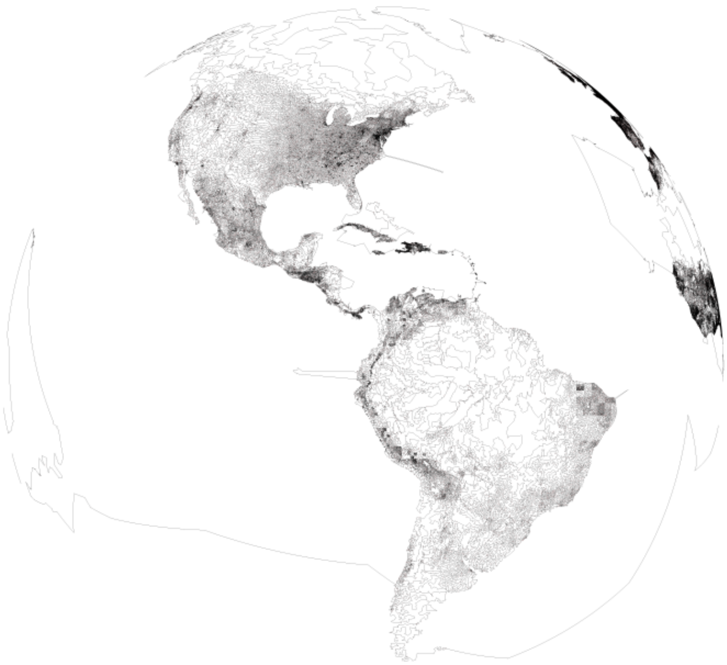
Reason: there are lots of routes.

For example, on a weekly schedule a route might consist of 20 legs.

If there are m legs in total, then there are up to m^{20} possible routes.

Traveling Salesman Problem

Most famous example of a *hard* problem:



Given n cities, determine the order in which to visit them so as to minimize the total travel distance.

Fixed Costs

$$c(x) = \begin{cases} 0 & \text{if } x = 0 \\ K + cx & \text{if } x > 0. \end{cases}$$

Equivalent to:

$$c(x) = Ky + cx$$

together with the following constraints:

$$\begin{aligned} x &\leq uy \\ x &\geq 0 \\ y &\in \{0, 1\}. \end{aligned}$$

where u is an upper bound on x.

Nonlinear Objective Functions

Nonlinear objective functions are sometimes approximated by piecewise linear functions.

Piecewise linear functions can be treated using techniques similar to the fixed cost method above.

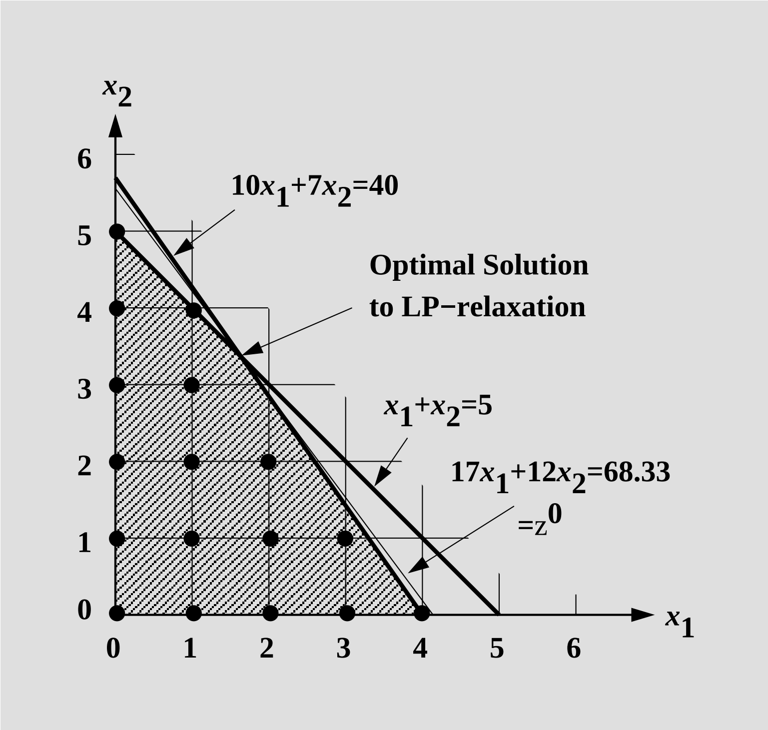
LP Relaxation

General Integer Programming Problem

$$\begin{array}{ll} \text{maximize} & c^T x \\ \text{subject to} & Ax \leq b \\ & x \geq 0 \\ & x \text{ has integer components.} \end{array}$$

Example

$$\begin{array}{lrcrcr} \text{maximize} & 17x_1 & + & 12x_2 & & \\ \text{subject to} & 10x_1 & + & 7x_2 & \leq & 40 \\ & x_1 & + & x_2 & \leq & 5 \\ & & & x_1,\ x_2 & \geq & 0 \\ & & & x_1,\ x_2 & & \text{integers.} \end{array}$$



Ignoring integrality constraints...
optimal solution is $(x_1, x_2) = (1.67, 3.33)$ with objective value 68.33.

Rounding to integers: $(2, 3)$ $\Longleftarrow$ infeasible.

Closest feasible: $(1, 3)$ $\Longleftarrow$ suboptimal.

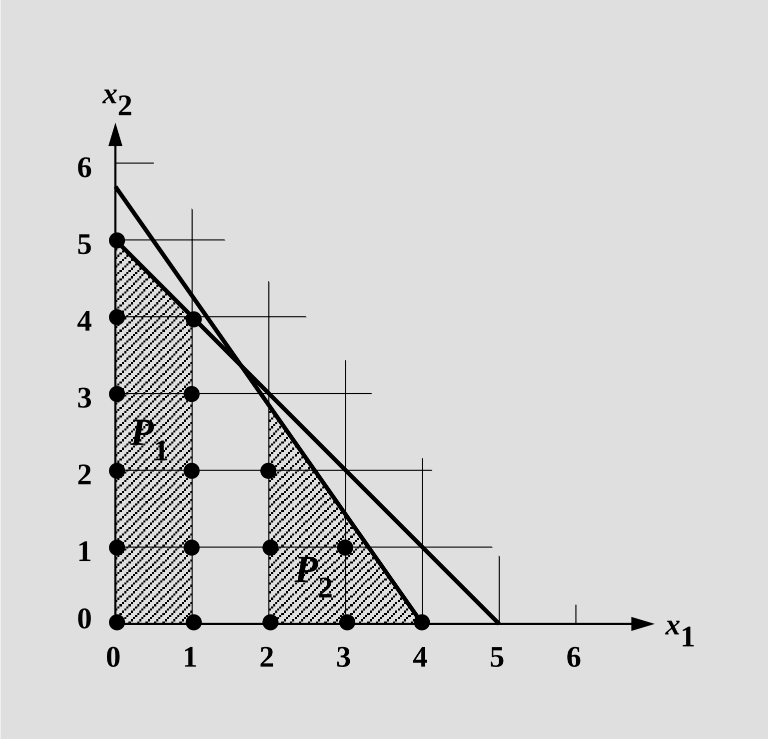
Branch-and-Bound

In LP relaxation, $x_1^* = 1.67$. Two possibilites:

- $x_1 \leq 1$
- $x_1 \geq 2$

Let

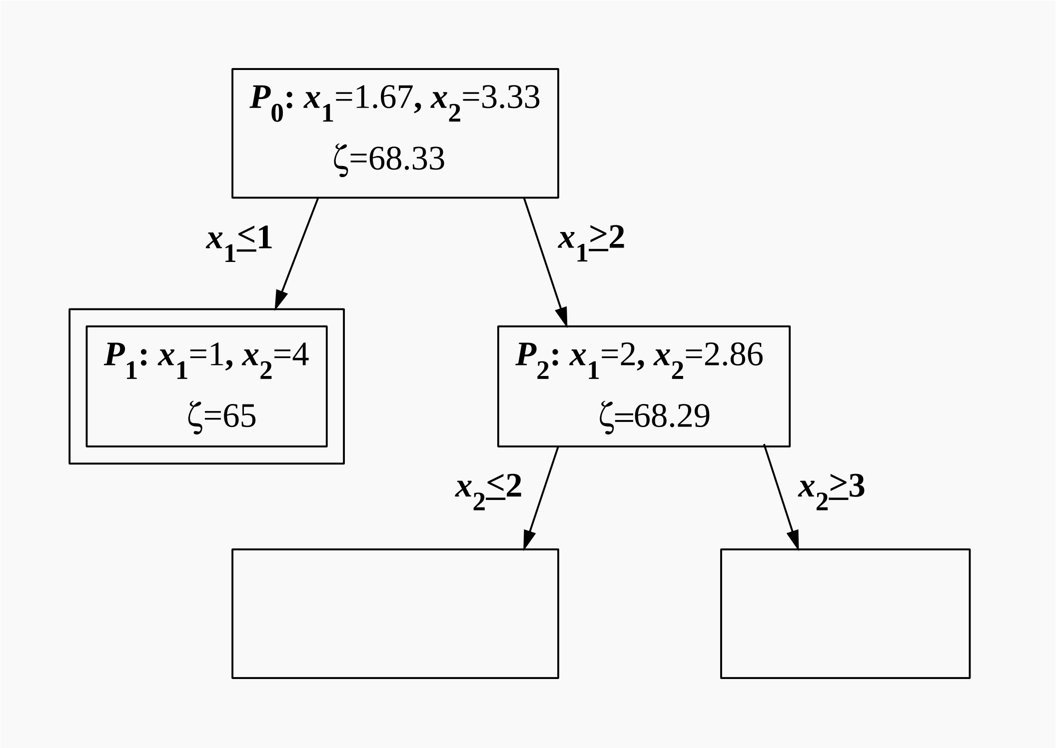
- P_1 = LP relaxation plus: $x_1 \leq 1$
- P_2 = LP relaxation plus: $x_1 \geq 2$



Optimal Solutions

- P_1: $(1, 4)$ $\Longleftarrow$ integer solution!
- P_2: $(2, 2.86)$

Enumeration Tree



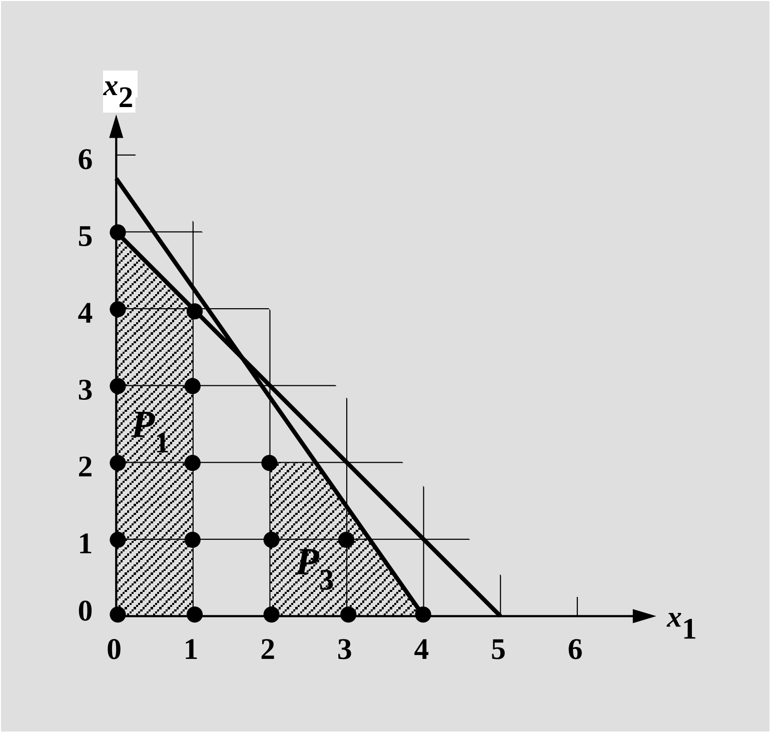
Double boxed node represents integer solution.

Integer solutions provide lower bounds on optimal integer solution.

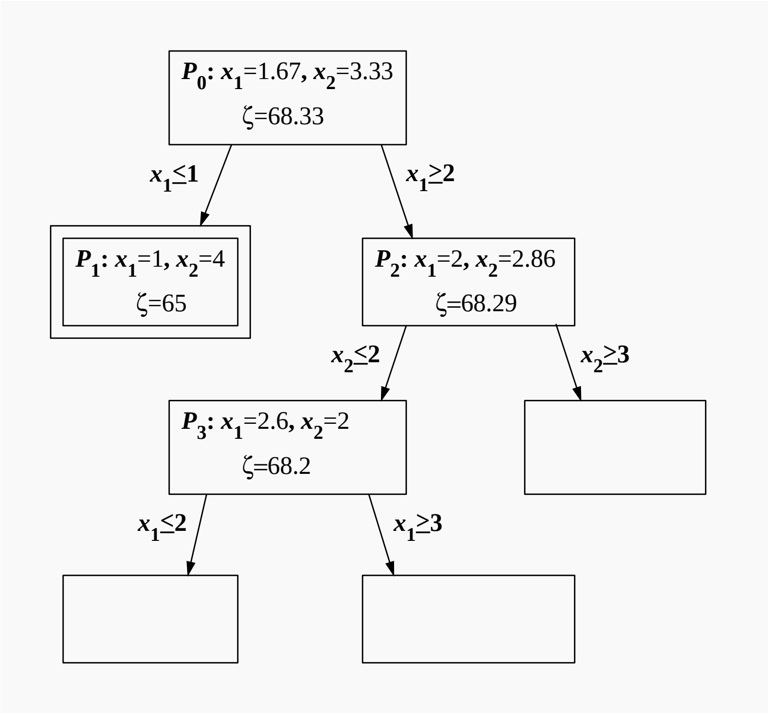
LP relaxations at each node provide upper bounds for the subtree below it.

Refinement of P_2 to P_3

Feasible Region:

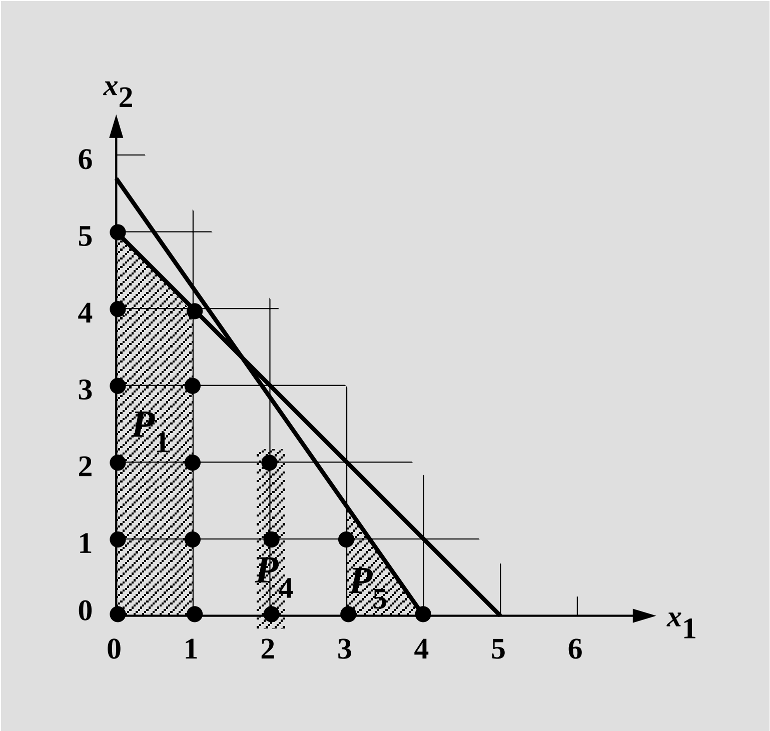


Enumeration Tree:

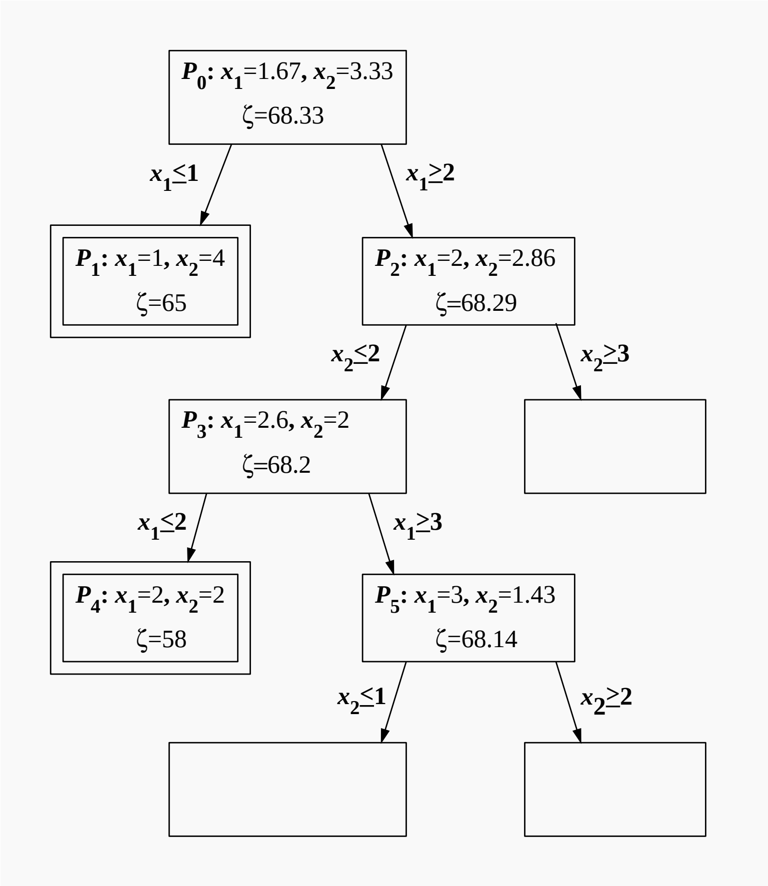


Splitting of P_3 into P_4 and P_5

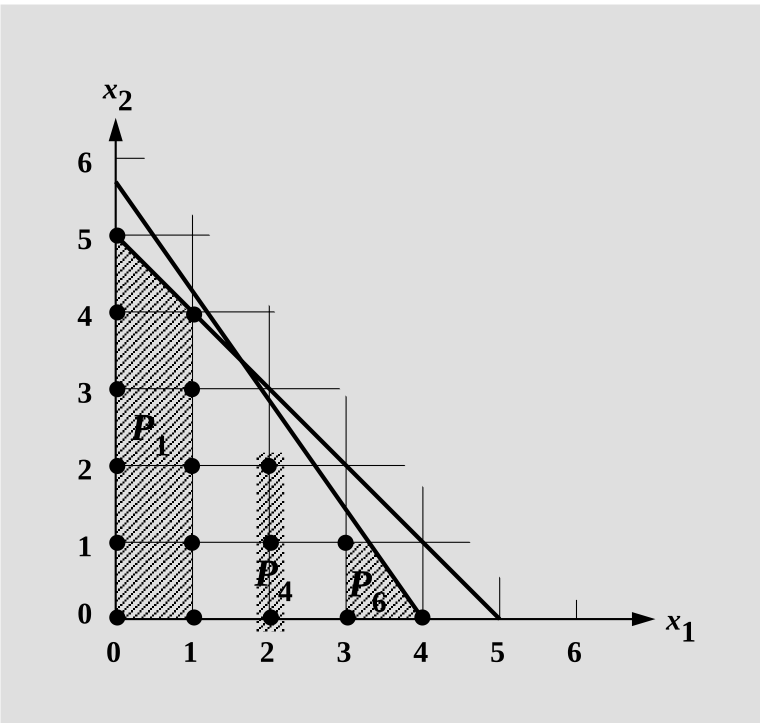
Enumeration Tree is Growing

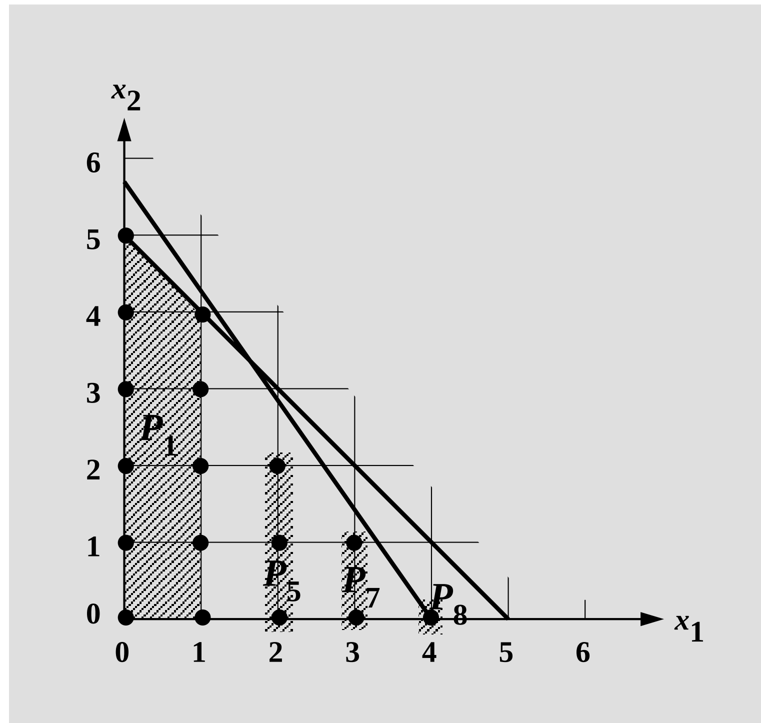


Enumeration Tree is Growing

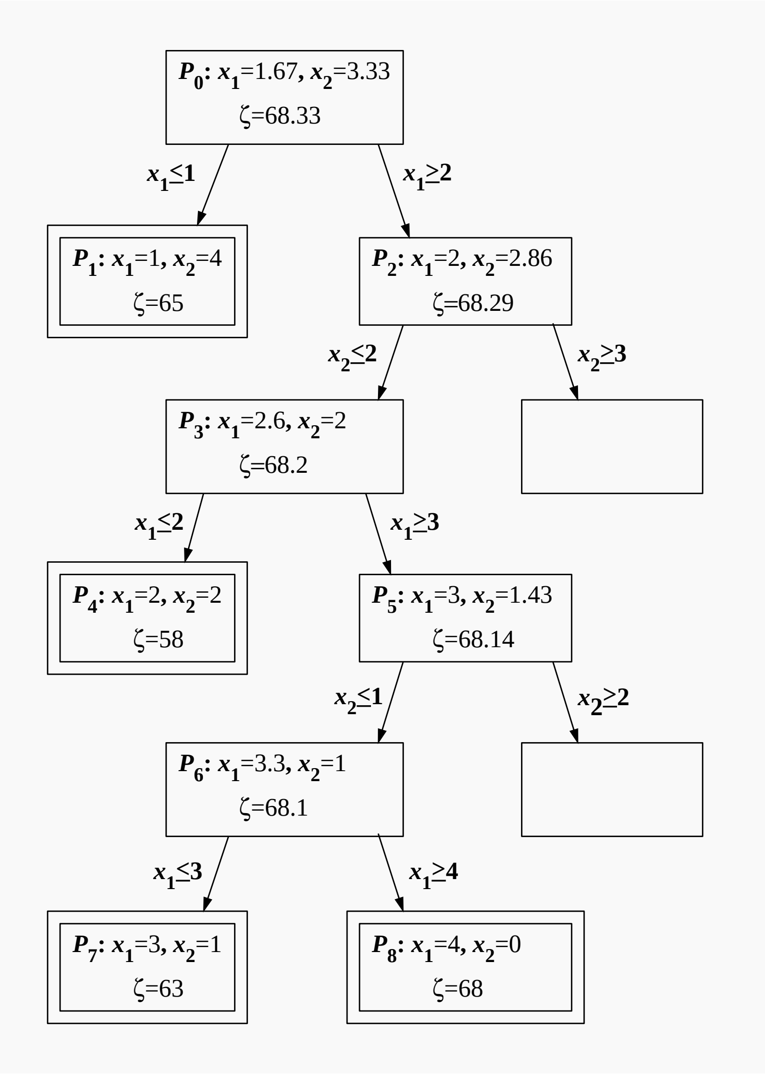


More Branching

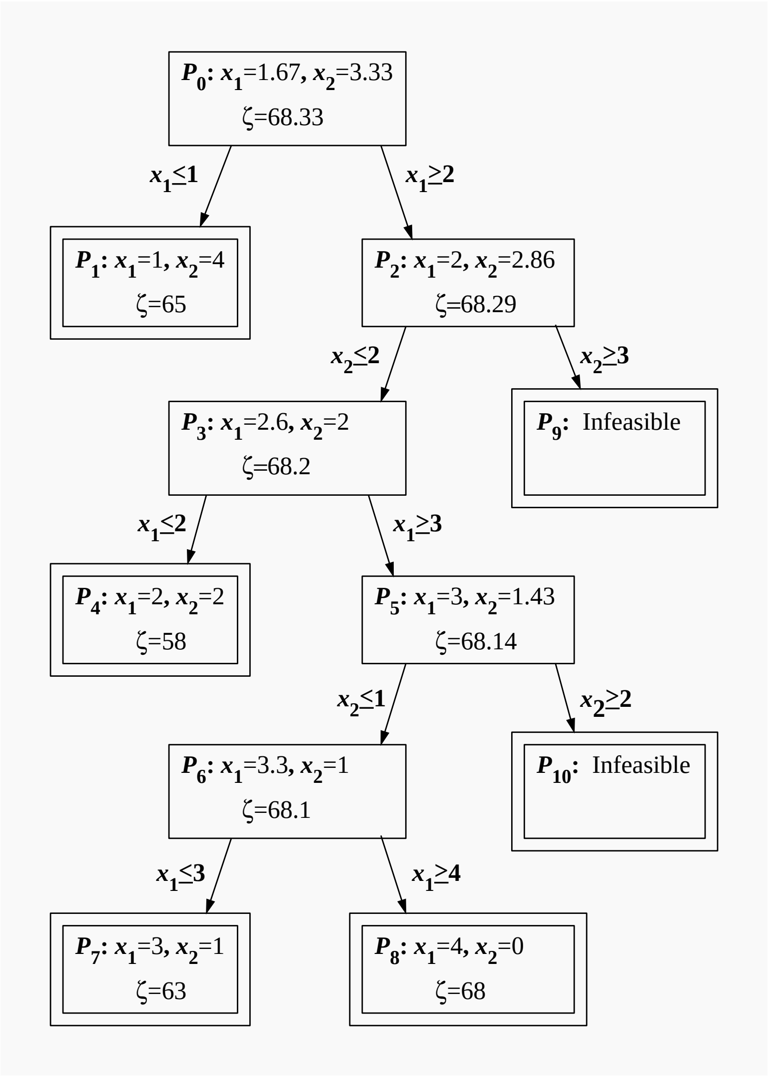




Enumeration Tree Still Growing



The Complete Enumeration Tree



Optimal solution: $(x_1, x_2) = (4, 0)$.