Integer Programming

$$\max \quad \zeta(x) = c^T x$$

$$\text{s.t.} \quad Ax \le b$$

$$x \ge 0, \qquad x_j \in \mathbb{Z}$$

Integer Programming

$$\max \quad \zeta(x) = \vec{c}^T x$$

$$\text{s.t.} \quad Ax \le b$$

$$x \ge 0, \quad x_j \in \mathbb{Z}$$

relaxed problem

Rk: Solving the problem without the int. constraints and then round up/down the solution does not work because

(1) rounded solution might not be feasible

(2) true opt. int. solution might be far from the rounded solution.

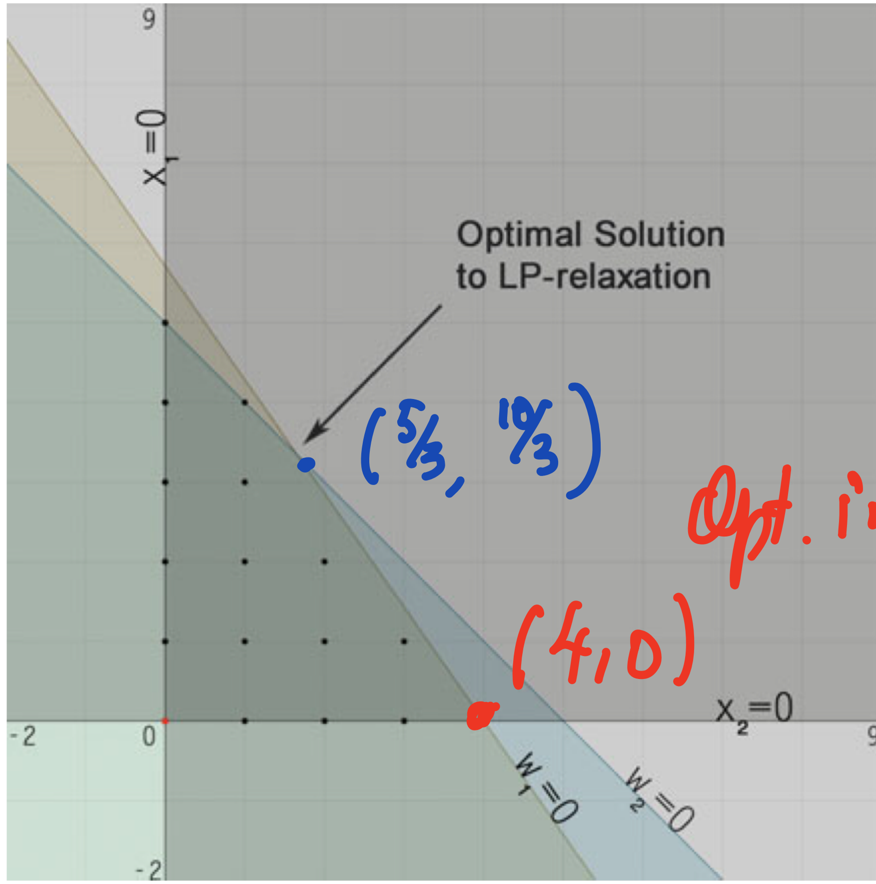
[V] p. 397

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

[V] p. 397

$$
\begin{array}{ll}
\text{maximize} & 17x_1 + 12x_2 \\
\text{subject to} & 10x_1 + \ 7x_2 \le 40 \\
 & x_1 + x_2 \le \ 5 \\
 & x_1,\ x_2 \ge \ 0 \\
 & \cancel{x_1,\ x_2 \text{ integers.}}
\end{array}
$$

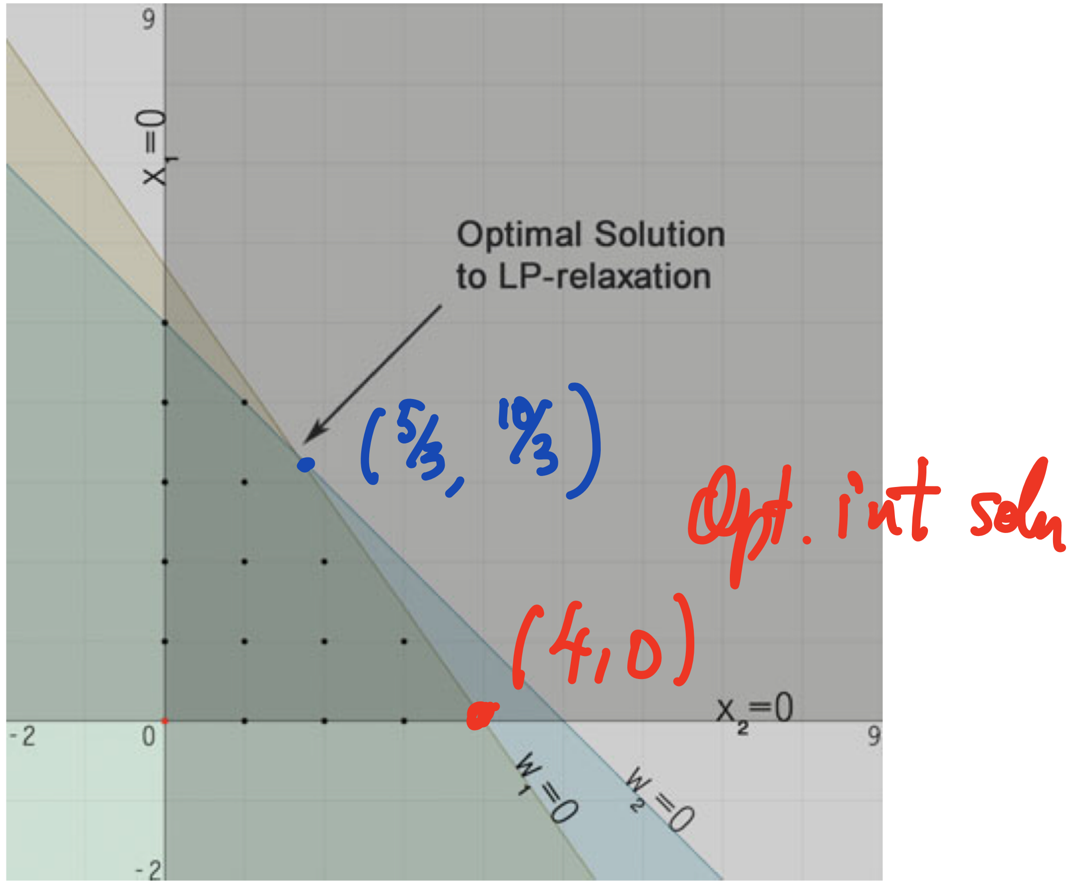
$x_1^* = \frac{5}{3}$, $x_2^* = \frac{10}{3}$



Opt. int soln

Branch-and-Bound

$$
\begin{array}{ll}
\text{maximize} & 17x_1 + 12x_2 \\
\text{subject to} & 10x_1 + 7x_2 \le 40 \\
& x_1 + x_2 \le 5 \\
& x_1,\ x_2 \ge 0 \\
& \cancel{x_1,\ x_2 \text{ integers.}}
\end{array}
$$



$x_1^* = \frac{5}{3}$, $\quad x_2^* = \frac{10}{3}$

$x_1 \le 1 \qquad x_1 \ge 2 \quad \leftarrow$ 2 sub-problems, with new constraints

Branch-and-Bound

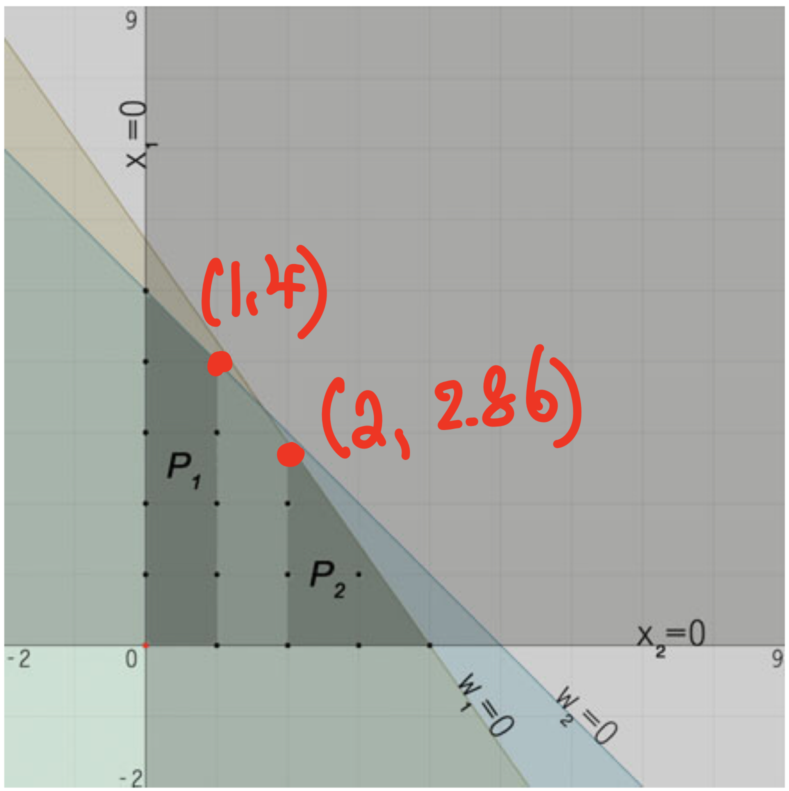


FIGURE 23.6. The feasible subregions formed by the first branch.

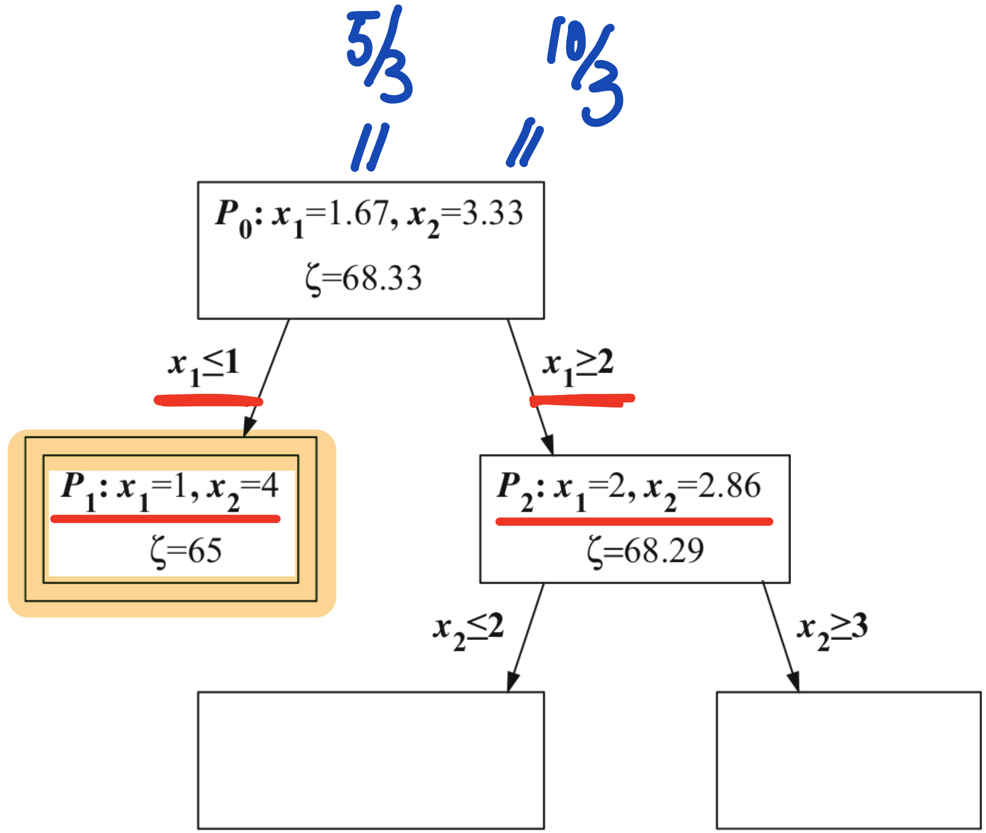


FIGURE 23.7. The beginnings of the enumeration tree.

Branch-and-Bound

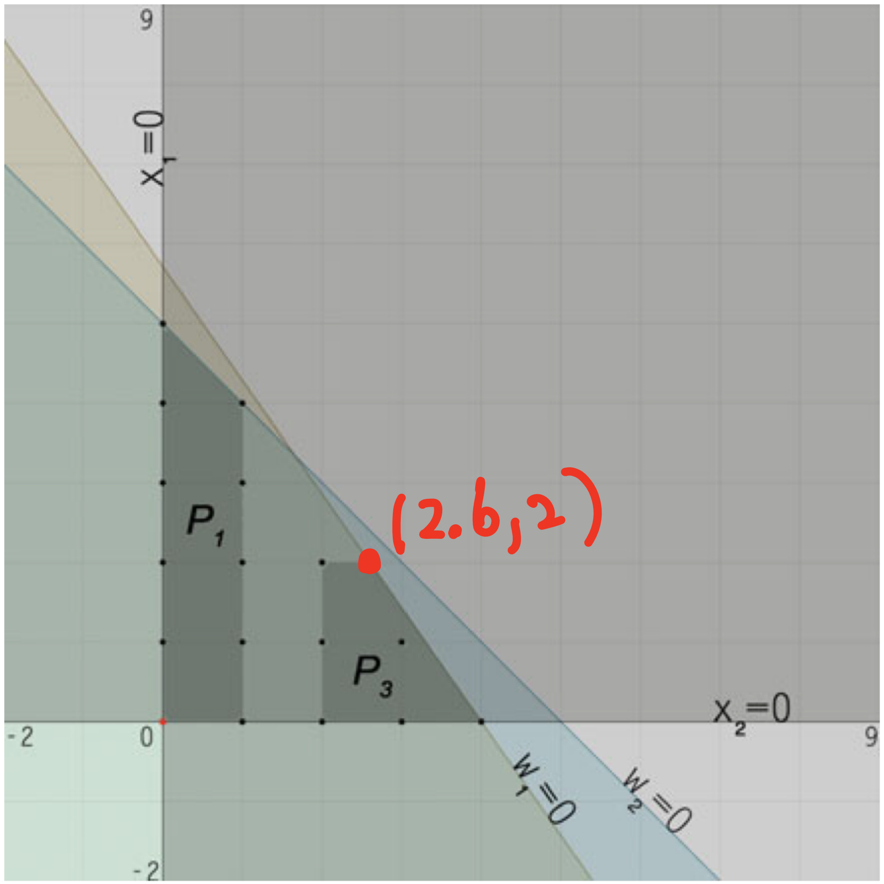


FIGURE 23.8. The refinement of P_2 to P_3.

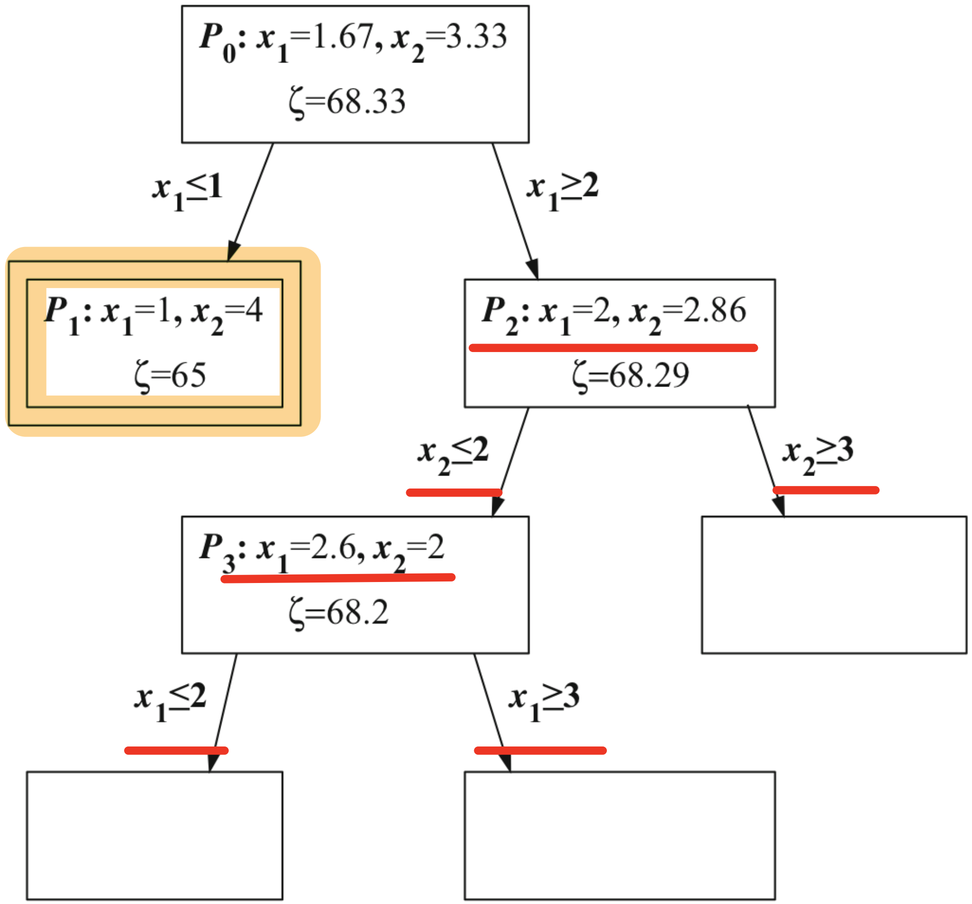


FIGURE 23.9. The enumeration tree after solving P_3.

Branch-and-Bound

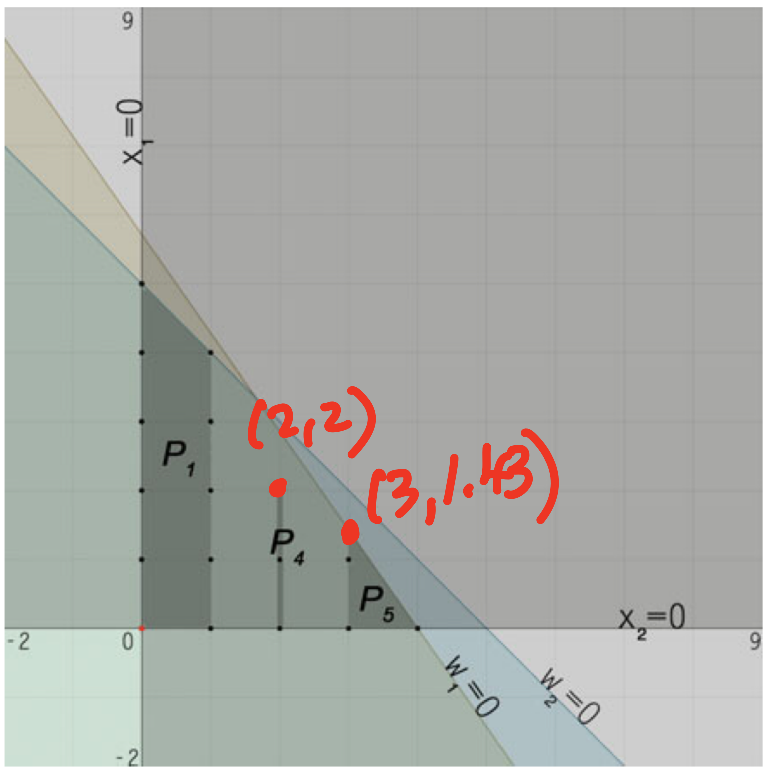


FIGURE 23.10. The refinement of P_3 to P_4 and P_5. Note that P_4 is just a short vertical line segment.

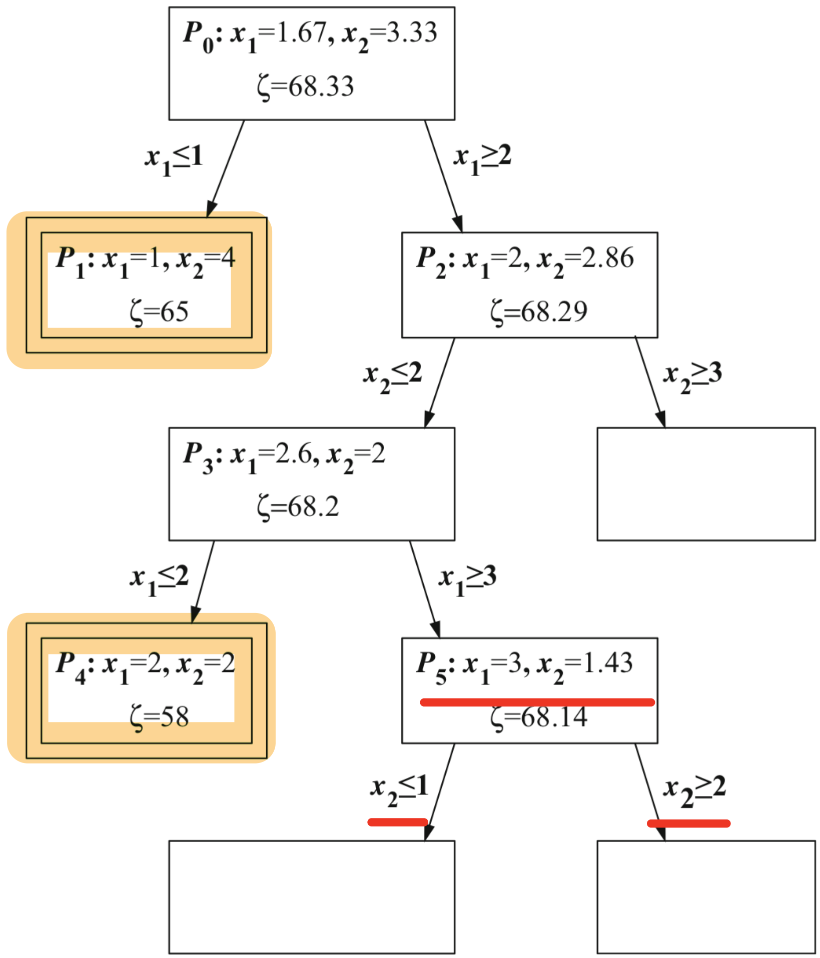


FIGURE 23.11. The enumeration tree after solving P_5. The double box around P_4 indicates that it is a leaf in the tree.

Branch-and-Bound

FIGURE 23.12. The refinement of P_5 to P_6.

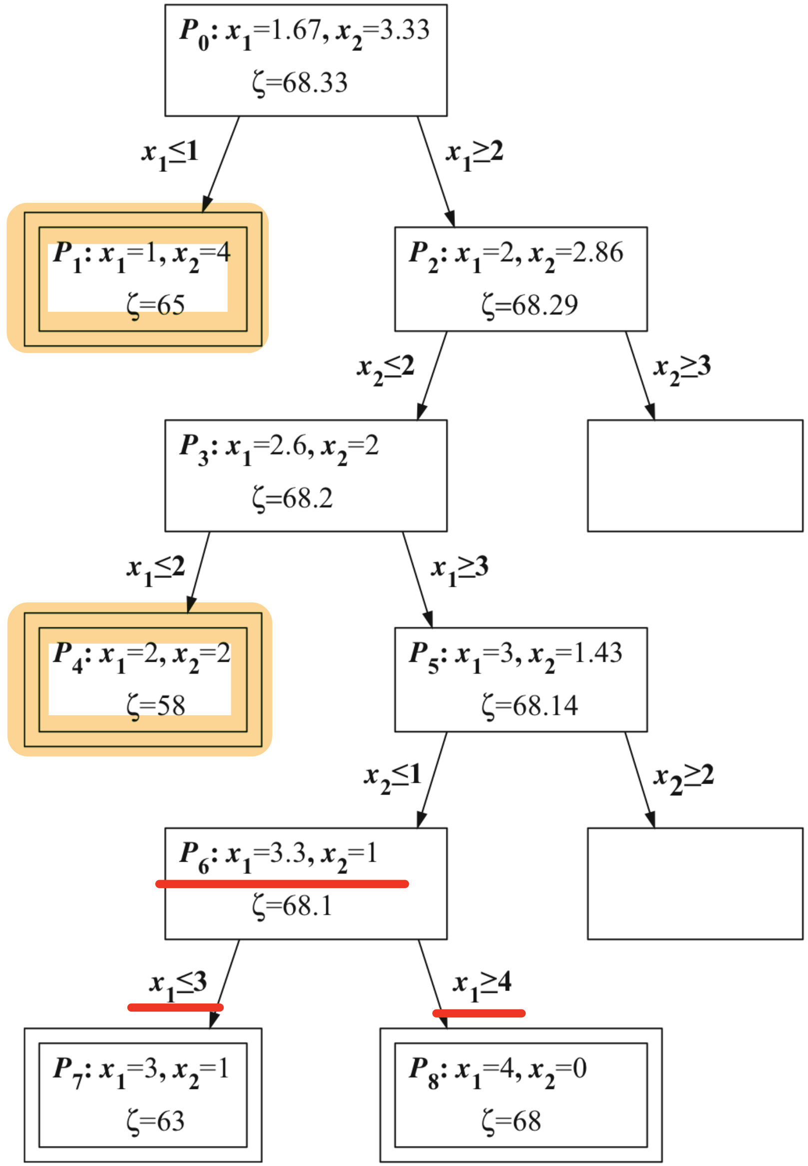


FIGURE 23.14. The enumeration tree after solving P_6, P_7, and P_8.

Branch-and-Bound

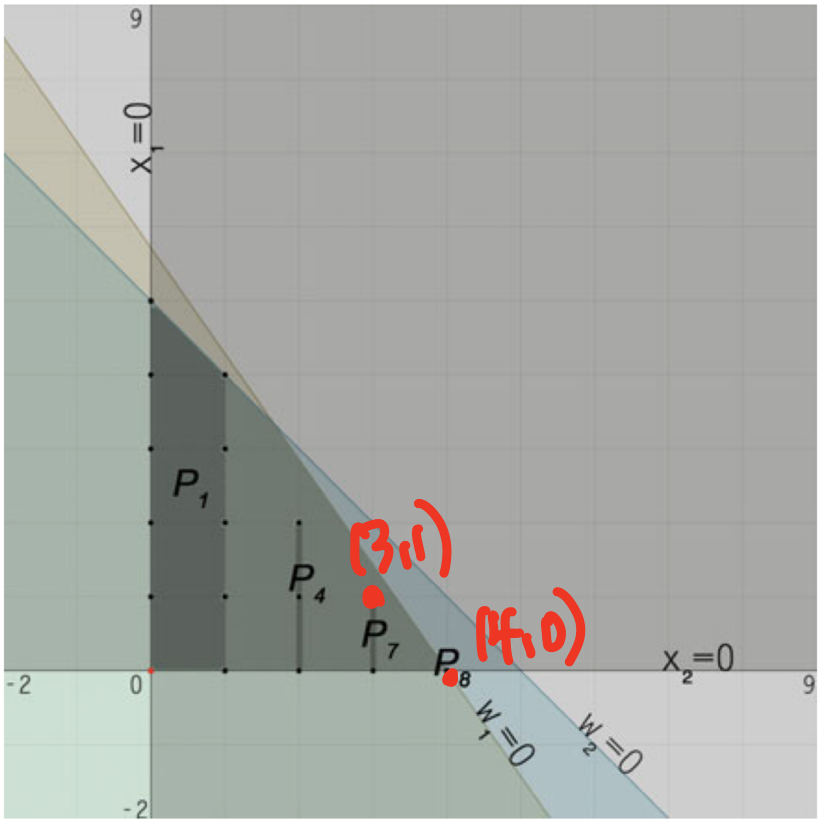


FIGURE 23.13. The refinement of P_6 to P_7 and P_8. Note that P_8 consists of just a single point: $(4, 0)$.

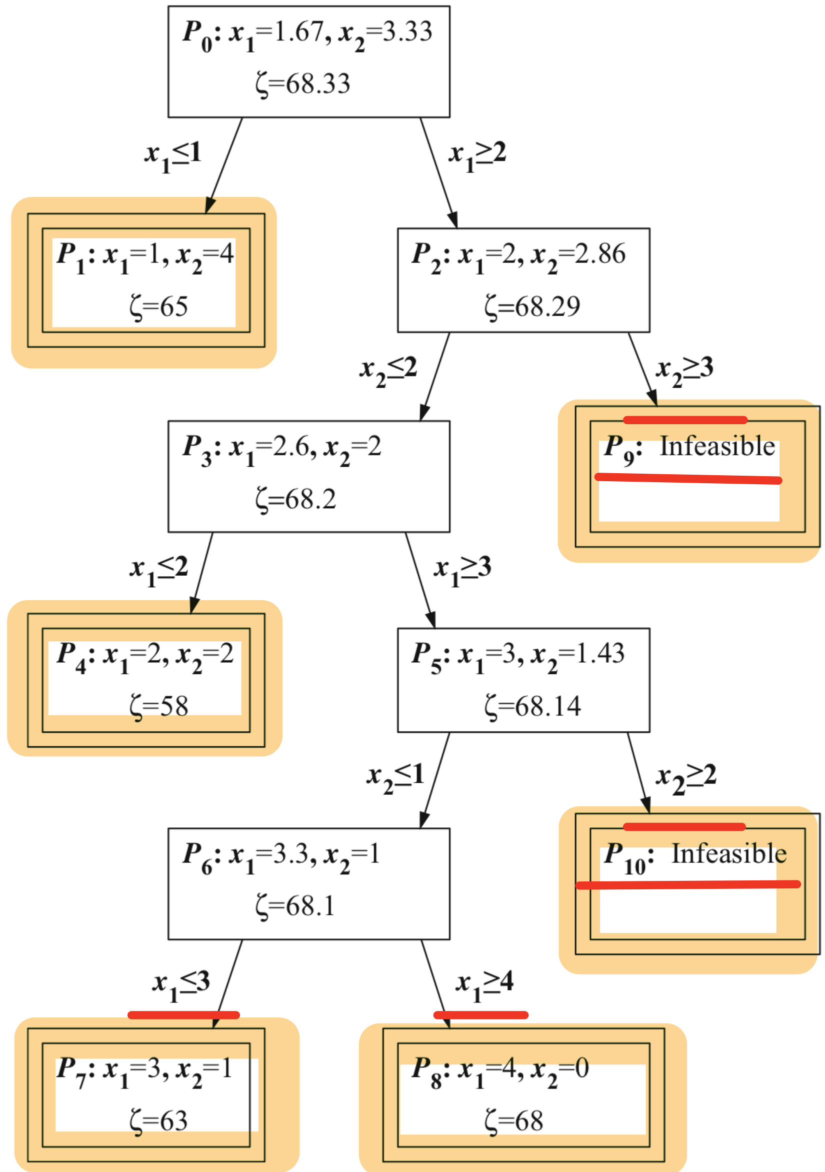


FIGURE 23.15. The complete enumeration tree.

Branch-and-Bound

It is better going deep – depth search
before going wide – breadth search

There are three reasons why depth-first search is generally the preferred order in which to fathom the enumeration tree. The first is based on the observation that most integer solutions lie deep in the tree. There are two advantages to finding integer feasible solutions early. The first is simply the fact that it is better to have a feasible solution than nothing in case one wishes to abort the solution process early. But more importantly, identifying a feasible integer solution can result in subsequent nodes of the enumeration tree being made into leaves simply because the optimal objective function associated with that node is lower than the best-so-far integer solution. Making such nodes into leaves is called *pruning* the tree and can account for tremendous gains in efficiency.

Branch-and-Bound

It is better going deep – depth search
before going wide – breadth search

2

A second reason to favor depth-first search is the simple fact that it is very easy to code the algorithm as a recursively defined function. This may seem trite, but one should not underestimate the value of code simplicity when implementing algorithms that are otherwise quite sophisticated, such as the one we are currently describing.

Branch-and-Bound

It is better going deep – depth search
before going wide – breadth search

3

The third reason to favor depth-first search is perhaps the most important. It is based on the observation that as one moves deeper in the enumeration tree, each subsequent linear programming problem is obtained from the preceding one by simply adding (or refining) an upper/lower bound on one specific variable. To see why this is an advantage, consider, for example, problem P_2, which is a refinement of P_0. The optimal dictionary for problem P_0 is recorded as

$$\begin{array}{rcrcrcr}
\zeta & = & \frac{205}{3} & - & \frac{5}{3}w_1 & - & \frac{1}{3}w_2 \\
\hline
x_1 & = & \frac{5}{3} & - & \frac{1}{3}w_1 & + & \frac{7}{3}w_2 \\
x_2 & = & \frac{10}{3} & + & \frac{1}{3}w_1 & - & \frac{10}{3}w_2
\end{array}$$

Problem P_2 is obtained from P_0 by adding the constraint that $x_1 \geq 2$. Introducing a variable, g_1, to stand for the difference between x_1 and this lower bound and using the dictionary above to write x_1 in terms of the nonbasic variables, we get

$$g_1 = x_1 - 2 = -\frac{1}{3} - \frac{1}{3}w_1 + \frac{7}{3}w_2.$$

Branch-and-Bound

It is better going deep – depth search
before going wide – breadth search

3

Therefore, we can use the following dictionary as a starting point for the solution of P_2:

$$
\begin{array}{rcrcrcr}
\zeta & = & \frac{205}{3} & - & \frac{5}{3}\,w_1 & - & \frac{1}{3}\,w_2 \\
\hline
x_1 & = & \frac{5}{3} & - & \frac{1}{3}\,w_1 & + & \frac{7}{3}\,w_2 \\
x_2 & = & \frac{10}{3} & + & \frac{1}{3}\,w_1 & - & \frac{10}{3}\,w_2 \\
g_1 & = & -\frac{1}{3} & - & \frac{1}{3}\,w_1 & + & \frac{7}{3}\,w_2
\end{array}
$$

← dual feasible

← new constraint.

This dictionary is dual feasible but primal infeasible. Therefore, the dual simplex method is likely to find a new optimal solution in very few iterations. According to

Gomory Cuts

$$\begin{array}{lrcl}
\text{maximize} & 8x_1 + 13x_2 & & \\
\text{subject to} & 10x_1 - \;\; 2x_2 & \leq & 32 \\
& 4x_1 + 10x_2 & \leq & 38 \\
& -2x_1 + 10x_2 & \leq & 29 \\
& x_1, \; x_2 & \geq & 0 \\
& x_1, \; x_2 & & \text{integers.}
\end{array}$$

Note:

A, b, X are integers and so are slacks

Gomory Cuts

$$\begin{array}{lrcl} \text{maximize} & 8x_1 + 13x_2 & & \\ \text{subject to} & 10x_1 - \;\; 2x_2 & \leq & 32 \\ & 4x_1 + 10x_2 & \leq & 38 \\ & -2x_1 + 10x_2 & \leq & 29 \\ & x_1,\; x_2 & \geq & 0 \\ \text{relaxed} & \cancel{x_1,\; x_2 \text{ integers}}. & & \end{array}$$

Optimal dictionary

$$\begin{array}{rcrcrcr} \zeta & = & \frac{179}{3} & - & \frac{7}{27}\,w_1 & - & \frac{73}{54}\,w_2 \\ \hline x_1 & = & \frac{11}{3} & - & \frac{5}{54}\,w_1 & - & \frac{1}{54}\,w_2 \\ x_2 & = & \frac{7}{3} & + & \frac{1}{27}\,w_1 & - & \frac{5}{54}\,w_2 \\ w_3 & = & 13 & - & \frac{5}{9}\,w_1 & + & \frac{8}{9}\,w_2 \end{array}$$

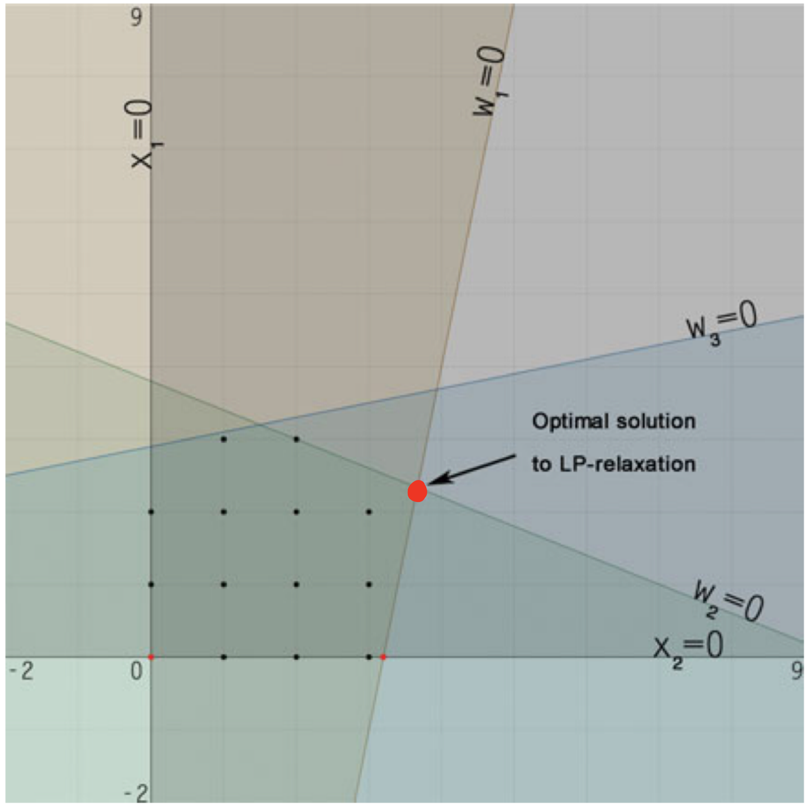


FIGURE 23.16. The integer program given in Equation (23.5). The red dots mark the points visited by the simplex method applied to the LP-relaxation.

Gomory Cuts

$$x_1 = \frac{11}{3} - \frac{5}{54} w_1 - \frac{5}{54} w_2 \implies \underbrace{x_1 + 0 w_1 + 0 w_2 \leq 3}_{\text{new constraint}}$$

Gomory Cuts

$$x_1 = \frac{11}{3} - \frac{5}{54}w_1 - \frac{5}{54}w_2 \Rightarrow \underbrace{x_1 + 0w_1 + 0w_2 \leq 3}_{\text{new constraint}}$$

$$x_1 + \frac{5}{54}w_1 + \frac{5}{54}w_2 = \frac{11}{3}$$

round down

$$\underbrace{x_1 + 0w_1 + 0w_2}_{\text{integer}} = \underbrace{\frac{11}{3} - \frac{5}{54}w_1 - \frac{5}{54}w_2}$$

$\Rightarrow$ integer, $\leq \frac{11}{3}$

$\Rightarrow$ integer, ≤ 3

Gomory Cuts

$$x_1 = \frac{11}{3} - \frac{5}{54}w_1 - \frac{5}{54}w_2 \Rightarrow \underbrace{x_1 + 0w_1 + 0w_2 \le 3}_{\text{new constraint}}$$

$$x_1 + \frac{5}{54}w_1 + \frac{5}{54}w_2 = \frac{11}{3}$$

round down

$$\frac{11}{3} - \frac{5}{54}w_1 - \frac{5}{54}w_2 \le 3$$

$$-\frac{5}{54}w_1 - \frac{5}{54}w_2 \le -\frac{2}{3}$$

new slack var: $\boxed{w_4 = -\frac{2}{3} + \frac{5}{54}w_1 + \frac{5}{54}w_2}$

Gomory Cuts

$$
\begin{array}{rcrcrcrcr}
\zeta & = & \frac{179}{3} & - & \frac{7}{27} & w_1 & - & \frac{73}{54} & w_2 \\
\hline
x_1 & = & \frac{11}{3} & - & \frac{5}{54} & w_1 & - & \frac{1}{54} & w_2 \\
x_2 & = & \frac{7}{3} & + & \frac{1}{27} & w_1 & - & \frac{5}{54} & w_2 \\
w_3 & = & 13 & - & \frac{5}{9} & w_1 & + & \frac{8}{9} & w_2 \\
w_4 & = & -\frac{2}{3} & + & \frac{5}{54} & w_1 & + & \frac{1}{54} & w_2
\end{array}
$$

new constraint

new boundary

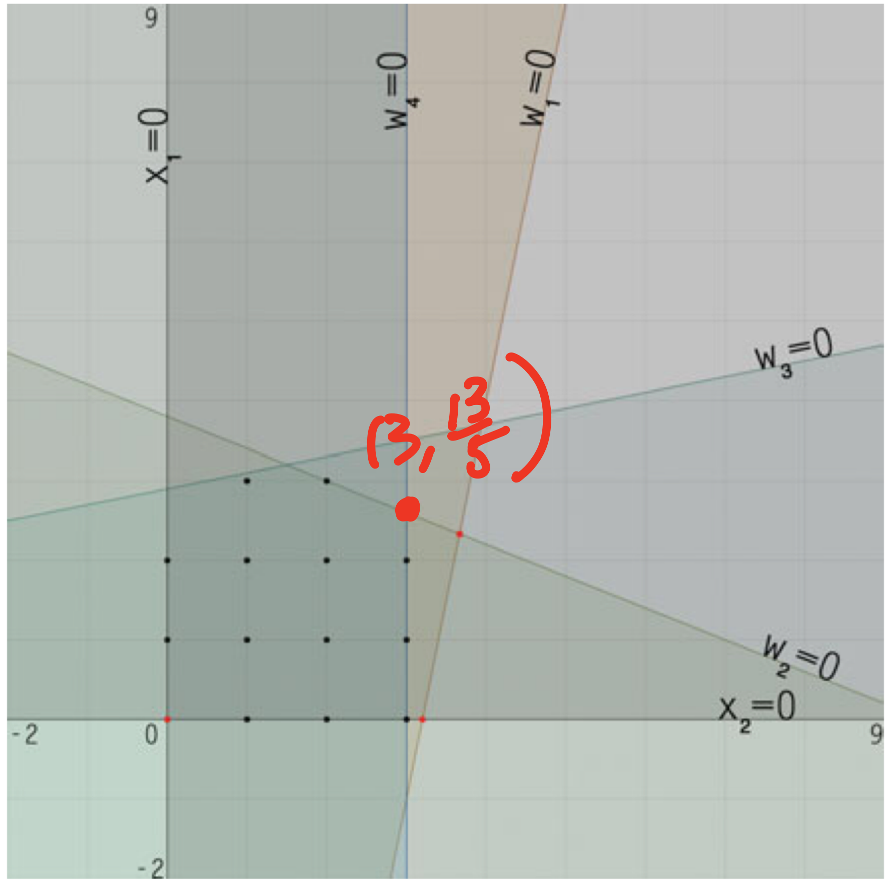


FIGURE 23.17. The integer program with our first new Gomory cut constraint added.

Opt. dict.

$$
\begin{array}{rcrcrcrcr}
\zeta & = & \frac{289}{5} & - & \frac{14}{5} & w_4 & - & \frac{13}{10} & w_2 \\
\hline
x_1 & = & 3 & - & & w_4 & & & \\
x_2 & = & \frac{13}{5} & + & \frac{2}{5} & w_4 & - & \frac{1}{10} & w_2 \\
w_3 & = & 9 & - & 6 & w_4 & + & 1 & w_2 \\
w_1 & = & \frac{36}{5} & + & \frac{54}{5} & w_4 & - & \frac{1}{5} & w_2
\end{array}
$$

Gomory Cuts

Opt. dict.

$$
\begin{array}{rcrcrlcrl}
\zeta & = & \frac{289}{5} & - & \frac{14}{5} & w_4 & - & \frac{13}{10} & w_2 \\
\hline
x_1 & = & 3 & - & & w_4 & & & \\
x_2 & = & \frac{13}{5} & + & \frac{2}{5} & w_4 & - & \frac{1}{10} & w_2 \\
w_3 & = & 9 & - & 6 & w_4 & + & 1 & w_2 \\
w_1 & = & \frac{36}{5} & + & \frac{54}{5} & w_4 & - & \frac{1}{5} & w_2
\end{array}
$$

$$x_2 - \frac{2}{5} w_4 + \frac{1}{10} w_2 = \frac{13}{5}$$

$$\underbrace{x_2 - w_4}_{\text{int.}} = \underbrace{\frac{13}{5} - \frac{3}{5} w_4 - \frac{1}{10} w_2}_{\text{int.} \le \frac{13}{5} \implies \text{int.} \le 2}$$

Gomory Cuts

Opt. dict.

$$
\begin{array}{rcrcrlcrl}
\zeta & = & \frac{289}{5} & - & \frac{14}{5} & w_4 & - & \frac{13}{10} & w_2 \\
\hline
x_1 & = & 3 & - & & w_4 & & & \\
x_2 & = & \frac{13}{5} & + & \frac{2}{5} & w_4 & - & \frac{1}{10} & w_2 \\
w_3 & = & 9 & - & 6 & w_4 & + & 1 & w_2 \\
w_1 & = & \frac{36}{5} & + & \frac{54}{5} & w_4 & - & \frac{1}{5} & w_2
\end{array}
$$

$$x_2 - \frac{2}{5} w_4 + \frac{1}{10} w_2 = \frac{13}{5}$$

$$\underbrace{x_2 - w_4}_{\text{int.}} = \underbrace{\frac{13}{5} - \frac{3}{5} w_4 - \frac{1}{10} w_2}_{\text{int.} \leq \frac{13}{5} \implies \text{int.} \leq 2}$$

$$\boxed{\begin{array}{l}
\frac{13}{5} - \frac{3}{5} w_4 - \frac{1}{10} w_2 \leq 2 \\
\text{New slack:} \\
w_5 = 2 - \left(\frac{13}{5} - \frac{3}{5} w_4 - \frac{1}{10} w_2\right) \\
\quad = -\frac{3}{5} + \frac{3}{5} w_4 + \frac{1}{10} w_2
\end{array}}$$

Gomory Cuts

$$
\begin{array}{rcl}
\zeta & = & \frac{289}{5} - \frac{14}{5} w_4 - \frac{13}{10} w_2 \\
\hline
x_1 & = & 3 - w_4 \\
x_2 & = & \frac{13}{5} + \frac{2}{5} w_4 - \frac{1}{10} w_2 \\
w_3 & = & 9 - 6 w_4 + 1 w_2 \\
w_1 & = & \frac{36}{5} + \frac{54}{5} w_4 - \frac{1}{5} w_2 \\
w_5 & = & -\frac{3}{5} + \frac{3}{5} w_4 + \frac{1}{10} w_2
\end{array}
$$

new slack

Opt. soln (int.)

$$
\begin{array}{rcl}
\zeta & = & 55 - \frac{14}{3} w_5 - \frac{5}{6} w_2 \\
\hline
x_1 & = & 2 - \frac{5}{3} w_5 + \frac{1}{6} w_2 \\
x_2 & = & 3 + \frac{2}{3} w_5 - \frac{1}{6} w_2 \\
w_3 & = & 3 - 10 w_5 + 2 w_2 \\
w_1 & = & 18 + 18 w_5 - 2 w_2 \\
w_4 & = & 1 + \frac{5}{3} w_5 - \frac{1}{6} w_2
\end{array}
$$

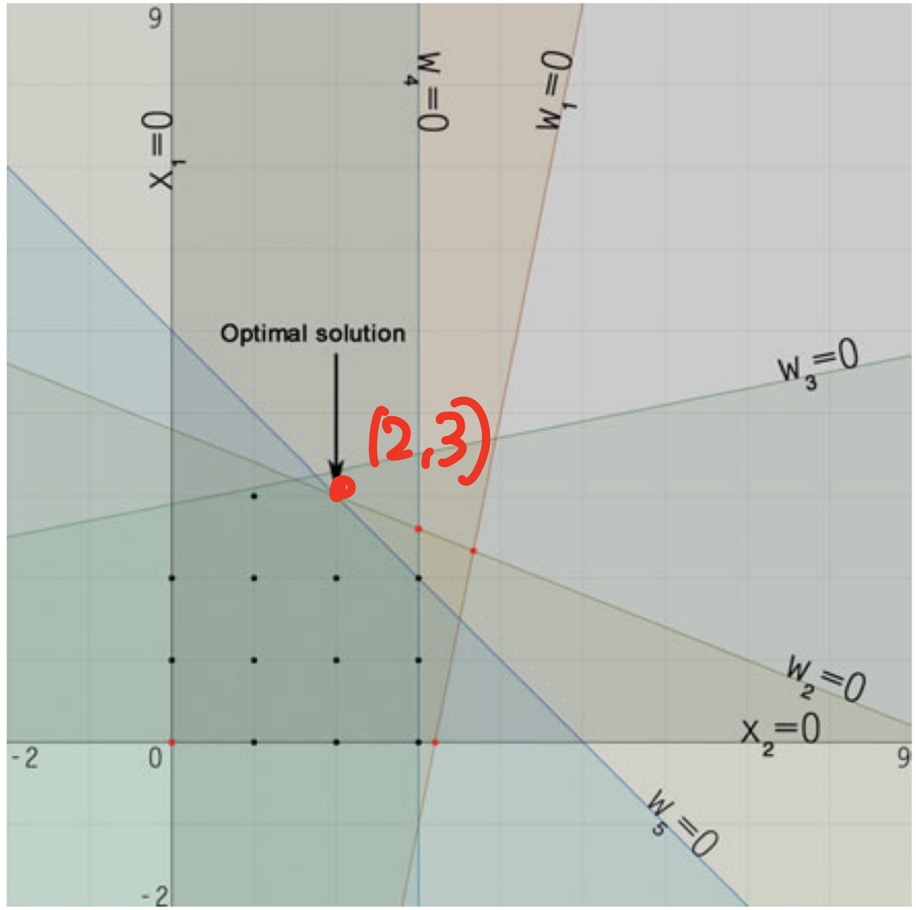


FIGURE 23.18. The integer program with two Gomory cuts added. This time the optimal solution to the LP-relaxation is an all integer solution.