MA427 – Mathematical Optimisation
Part II
Integer Programming

Giacomo Zambelli
g.zambelli@lse.ac.uk

Department of Mathematics
London School of Economics and Political Science
2018/19

Contents

Chapter 1

Integer programming problems and branch-and-bound

A *Integer Linear Programming Problem* is a linear programming problem in which some variables are required to take integer values. We can therefore write any integer programming problem in the form

$$\begin{aligned} z_I = \max c^\top x \\ Ax \le b \\ x \ge 0 \\ x_i \in \mathbb{Z}, \qquad i \in I. \end{aligned} \tag{1.1}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ and $I \subseteq \{1,\dots,n\}$ is the set of indices of the *integer variables.* The other variables, namely the variables x_i, $i \notin I$, are referred to as *continuous variables.* If the problem has both integer and continuous variables, then it is said a *Mixed-integer Linear Programming Problem* (MILP). If all variables are integer, the problem is said a *Pure Integer Linear Programming Problem* (ILP).

The set

$$X = \{x \in \mathbb{R}^n \mid Ax \le b,\ x \ge 0,\ x_i \in \mathbb{Z} \text{ for all } i \in I\}$$

is the *feasible region* of the problem.

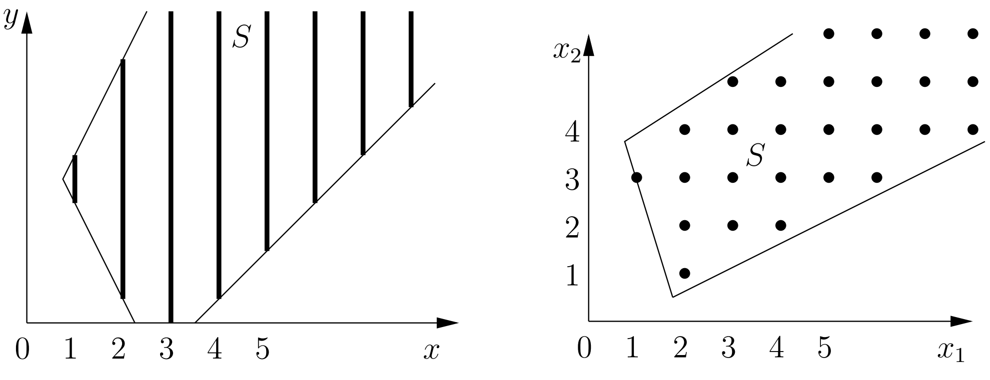


Figure 1.1: A mixed integer linear set and a pure integer linear set.

There is a natural linear programming problem associated with (1.1), namely the LP problem

$$\begin{aligned} z_L = \max c^\top x \\ Ax \le b \\ x \ge 0 \end{aligned} \tag{1.2}$$

The LP problem (1.2) is said the *linear relaxation* of (1.1).

Note the easy (but useful) fact:

$$z_I \le z_L. \tag{1.3}$$

Indeed, if x^I is the optimal solution of (1.1) and x^L is the optimal solution of (1.2), then x^I is feasible for (1.2),thus $z_I = c^\top x^I \leq c^\top x^L = z_L$.

Note in particular that, if we find a feasible solution $\bar{x}$ for the MILP (1.1) such that $c^\top \bar{x} = z_L$, we can conclude that $\bar{x}$ is an optimal solution for the MILP problem (1.1), and that $z_L = z_I$.

However, in general z_I might be different from z_L (and in practice it almost always is...). Thus solving MILP problems is more difficult than solving linear programming problem. There are two main approaches to solve MILP problems: branch-and-bound and cutting planes. We describe the first approach in the next section, and discuss cutting planes in Chapter 4.

1.1 Branch and Bound

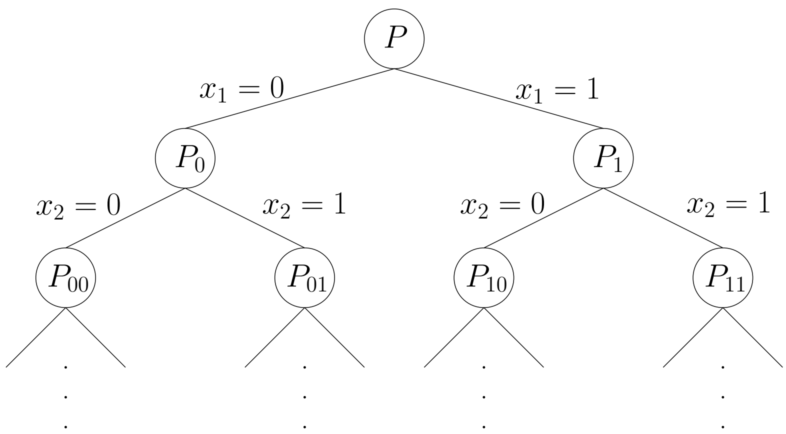
Branch and Bound is the first general method for solving integer programming problems, and it is the backbone of modern integer programming solvers. [1]

Given a partition of the feasible region X into sets $X_1, \ldots, X_n$, let $z_I^{(k)} = \max\{c^\top x \,|\, x \in X_k\}$. Then

$$z_I = \max_{k=1,\ldots,n} z_I^{(k)}.$$

Thus the branch-and-bound method partitions the set X into smaller subsets, and solves the subproblem $\max c^\top x$ on each of the subproblems.

For example, if we consider a problem P with n variables, all binary (that is, the variables are allowed to take value only $0, 1$), then we could consider the two subproblems P_0 and P_1 where the variable x_1 is fixed to 0 or to 1. To solve P_0 and P_1, we could fix in each problem the variable x_2 to either 0 or 1, thus getting two subproblems for each of P_0 and P_1. We could thus proceed until we have exhausted all possibilities.



Note that, if we have n variable, we will create 2^n subproblems if we enumerate them all. This makes using complete enumeration computationally infeasible: even if we could enumerate each of the 2^n subproblems in one microsecond (i.e. one millionth of a second), already enumerating all subproblems for $n = 50$ would require almost 36 years, for $n = 60$ it would require more than 37500 years, while for $n = 70$ it would require well beyond 37 million years.

To avoid enumerating too many subproblems, the Branch and Bound method seeks to exploit linear programming to obtain bounds on the optimal value of each subproblem, thus considering only the subproblems where an optimal solution might occur. To expose the method we resort to an example.

[1] A historical note: the branch-and-bound method was developed at the London School of Economics by Ailsa H. Land and Alison G. Doig, and was presented in the paper "An automatic method of solving discrete programming problems" published on *Econometrica* in 1960.

Example.

Consider the problem (P_0):

$$\begin{array}{rrcl} z_I^0 = \max & 5x_1 + \frac{17}{4}x_2 & & \\ & x_1 + x_2 & \le & 5 \\ & 10x_1 + 6x_2 & \le & 45 \\ & x_1, x_2 & \ge & 0 \\ & x_1, x_2 & \in & \mathbb{Z} \end{array} \qquad (P_0)$$

The feasible region of (P_0) is depicted in the Figure 1.2.

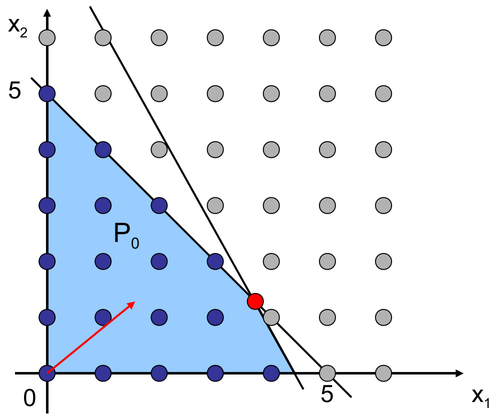


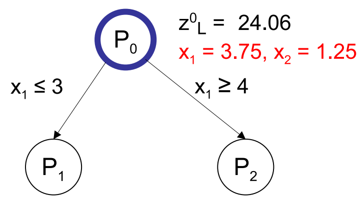
Figure 1.2: LP relaxation of P_0.

If we solve the linear relaxation of (P_0), the optimal solution is $x_1 = 3.75$, $x_2 = 1.75$, with value $z_L^0 = 24.06$.

Thus $z_I^0 \le 24.06$ is an upper-bound on the optimal value z_I^0 of (P_0). Now, since x_1 must be integer valued, any optimal solution of (P_0) must satisfy either $x_1 \le 3$ or $x_1 \ge 4$. Thus the optimal solution of (P_0) is the best among the optimal solutions of the problems (P_1) and (P_2) defined as follows:

$$\begin{array}{rrcl} z_I^1 = \max & 5x_1 + \frac{17}{4}x_2 & & \\ & x_1 + x_2 & \le & 5 \\ & 10x_1 + 6x_2 & \le & 45 \\ & x_1 & \le & 3 \\ & x_1, x_2 & \ge & 0 \\ & x_1, x_2 & \in & \mathbb{Z} \end{array} \quad (P_1) \quad , \qquad \begin{array}{rrcl} z_I^2 = \max & 5x_1 + \frac{17}{4}x_2 & & \\ & x_1 + x_2 & \le & 5 \\ & 10x_1 + 6x_2 & \le & 45 \\ & x_1 & \ge & 4 \\ & x_1, x_2 & \ge & 0 \\ & x_1, x_2 & \in & \mathbb{Z} \end{array} \quad (P_2)$$

We say that we *branched on variable* x_1. Note that the solution $(3.75, 1.75)$ is not feasible for the LP relaxation of any of the problems (P_1) or (P_2). We can represent graphically the subproblems in the so called *Branch-and-Bound tree*, in figure below.



The *active problems* are the leaves of the tree, in this case (P_1) and (P_2).

Consider now problem (P_1), represented in the Figure 1.3.
The optimal solution to the LP relaxation of (P_1) is $x_1 = 3$, $x_2 = 2$, with value $z_L^1 = 23.5$. Note that such solution is integer, thus it is also the optimal integer solution of (P_1). Thus we do not need to

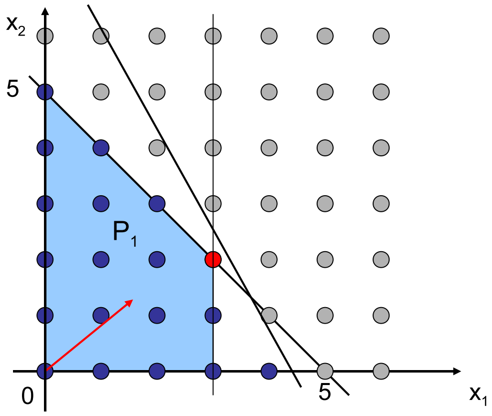
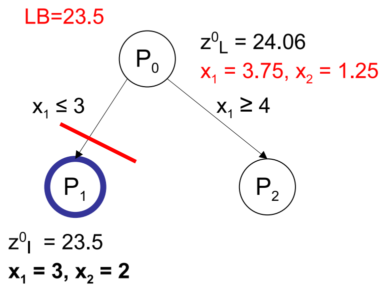


Figure 1.3: LP relaxation of P_1.

branch any further at node P_1, and thus we do not need to consider it anymore. We thus remove it from the tree, and say that (P_1) is *pruned by optimality.*

Note that, since every feasible solution to (P_1) is also feasible for P_0, we know that the optimal solution of P_0 cannot be worse than $(3, 2)$. Therefore $LB := 23.5$ is a lower-bound on the optimal value of (P_0), and $(3, 2)$ is said the *incumbent solution*, which is the best integer solution we have found thus far.

The branch-and-bound tree is now the following. The only active node is now is $(P2)$, which we



represent in Figure 1.4.

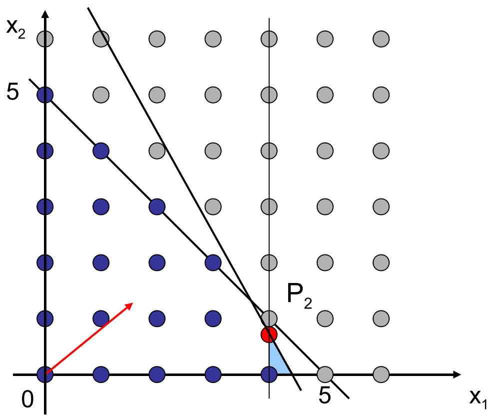
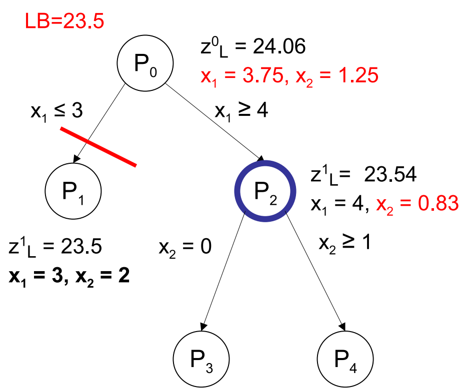


Figure 1.4: LP relaxation of P_2.

The optimal solution to the linear relaxation of (P_2) is $x_1 = 4$, $x_2 = 0.83$, with value $z_L^2 = 23.54$. Hence 23.54 is an upper-bound to the optimal value of (P_2). Note that $LB = 23.5 < 23.54$, thus (P_1) might have a better solution than the incumbent. Since $x_2 = 0.83$, we branch on x_2, obtaining the two following subproblems (P_3) and (P_4).

$$
\begin{array}{rrcl}
z_I^2 = \max & 5x_1 + \frac{17}{4}x_2 & & \\
& x_1 + x_2 & \leq & 5 \\
& 10x_1 + 6x_2 & \leq & 45 \\
& x_1 & \geq & 4 \\
& x_2 & \leq & 0 \\
& x_1, x_2 & \geq & 0 \\
& x_1, x_2 & \in & \mathbb{Z}
\end{array} \quad (P_3) \quad , \qquad
\begin{array}{rrcl}
z_I^2 = \max & 5x_1 + \frac{17}{4}x_2 & & \\
& x_1 + x_2 & \leq & 5 \\
& 10x_1 + 6x_2 & \leq & 45 \\
& x_1 & \geq & 4 \\
& x_2 & \geq & 1 \\
& x_1, x_2 & \geq & 0 \\
& x_1, x_2 & \in & \mathbb{Z}
\end{array} \quad (P_4)
$$

The branch-and-bound tree is the following



The active nodes are now $(P3)$ and (P_4). If we solve the LP relaxation of (P_3) (in Figure 1.5),

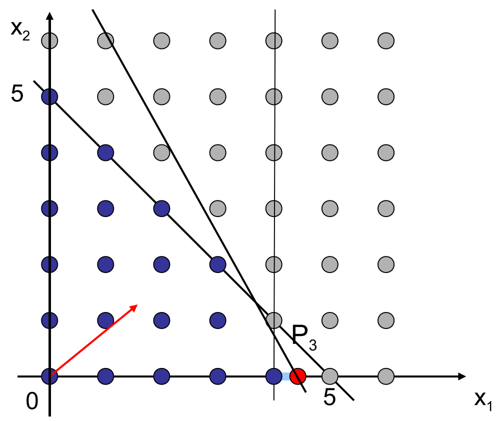
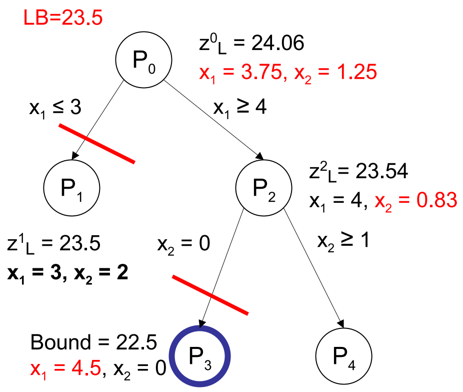


Figure 1.5: LP relaxation of P_3.

we obtain the optimal solution $x_1 = 4.5$, $x_2 = 0$, with value $z_L^3 = 22.5$. This shows that the optimal (integer) solution of (P_3) has value at most 22.5, but since this is worse than the value 23.5 of the incumbent solution, we do not need to explore any further the feasible region of (P_3). Hence we *prune node* (P_3) *by bound*. The branch-and-bound tree now has only one active node, namely (P_4).

If we solve the linear relaxation of (P_4), we determine that it is infeasible. In particular, (P_4) has no integer solution.



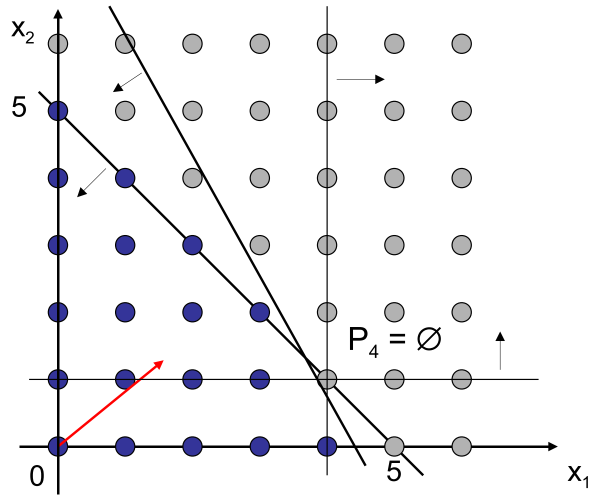
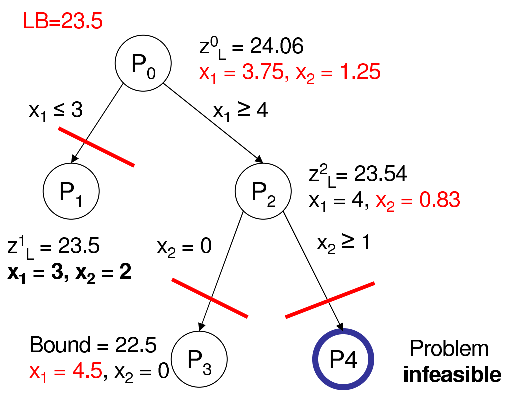


Figure 1.6: LP relaxation of P_4.

Thus we can *prune node* (P_4) *by infeasibility*. The branch-and-bound tree at this point has no active node, thus the incumbent solution $(3, 2)$ is the optimum.



Next we expose formally the branch-and-bound algorithm. Consider the mixed-integer linear programming problem (P_0)

$$\begin{aligned} z_I = \max c^\top x \\ Ax \leq b \\ x \geq 0 \\ x_i \in \mathbb{Z}, \qquad i \in I. \end{aligned}$$

where I is the set of indices of the integer variables.

The algorithm will maintain a lower-bound LB on the optimum value, and an incumbent solution x^* which is the best optimal solution found so far. (in particular $x_i^* \in \mathbb{Z}$ for every $i \in I$, and $c^\top x^* = LB$). We also maintain a branch-and-bound tree $\mathcal{T}$ whose leaves are the *active nodes*.

Branch-and-Bound method

Initialization: $\mathcal{T} := \{(P_0)\}$, $\ell := 0$, $LB := -\infty$, x^* undefined;

1. If there is no active node, then return the incumbent solution x^* as the optimum, STOP;
2. Else, choose an active node (P) in $\mathcal{T}$.
3. Solve the linear relaxation of (P).
   (a) **Pruning by infeasibility**: If the linear relaxation of (P) is infeasible, then prune (P); Else, let $\bar{x}$ be the optimum to the LP relaxation of (P).
   (b) **Pruning by bound**: If $c^\top \bar{x} \leq LB$, then (P) has no better solution than the incumbent x^*: prune (P);
   Else
   (c) **Pruning by optimality**: If $\bar{x}_i$ is integer for all $i \in I$, set $x^* := \bar{x}$ and $LB := c^\top \bar{x}$ and prune node (P).
   (d) **Branching on variable**: If none of (a), (b), (c) occurs, choose an integer variable x_h such that $\bar{x}_h \notin \mathbb{Z}$ and add in $\mathcal{T}$ as children of node (P) two new problems (P') and (P'') defined by
   $$(P') := (P) \cap \{x_h \leq \lfloor \bar{x}_h \rfloor\} \quad , \quad (P'') := (P) \cap \{x_h \geq \lceil \bar{x}_h \rceil\}.$$
4. Return to 1.

Table 1.1: Outline of the branch-and-bound method.

1.2 How to restart the simplex method after branching

Consider a mixed-integer linear programming problem, where I denotes the set of indices of integer variables.

Suppose that we are solving said problem with the Branch-and-Bound method, and that we are considering a subproblem (P) of the branch-and-bound tree. When we solve the LP relaxation of subproblem (P), we end up with an optimal basic feasible solution x^* relative to some optimal basis B (as usual we denote by N the set of indices of non-basic variables). The LP relaxation of the problem in tableau form with respect to the basis B will be of the form

$$\begin{array}{llrcl} \max & z & & & \\ & z & -\sum_{j \in N} \bar{c}_j x_j & = & z_B \\ & x_{\beta[i]} & +\sum_{j \in N} \bar{a}_{ij} x_j & = & \bar{b}_i, \quad i = 1, \ldots, m \\ & & x & \geq & 0. \end{array}$$

Since B is an optimal basis, all reduced costs are nonpositive, i.e. $\bar{c}_j \leq 0$, $j \in N$. The optimal solution x^* is defined by

$$\begin{array}{rcl} x^*_{\beta[i]} & = & \bar{b}_i, \quad i = 1, \ldots, m; \\ x^*_j & = & 0, \quad j \in N; \end{array}$$

Suppose that we next branch on some integer variable taking fractional value. Say, we branch on $x_{\beta[h]}$, where $h \in \{1, \ldots, m\}$ such that $\beta[h] \in I$ and $\bar{b}_h \notin \mathbb{Z}$. Thus we introduce two new sub-problems, namely

$$(P') := (P) \cap \{x_{\beta[h]} \le \lfloor \bar{b}_h \rfloor\} \quad , \quad (P'') := (P) \cap \{x_{\beta[h]} \ge \lceil \bar{b}_h \rceil\}.$$

Next we show how, when we need to resolve (P') or (P''), we do not need to start from scratch, but we can resolve them using the optimal basis B we found for (P).

To solve the LP relaxation of (P'), we introduce a new nonnegative continuous slack variable s for the new inequality $x_{\beta[h]} \le \lfloor \bar{b}_h \rfloor$, so that this can be expressed as

$$x_{\beta[h]} + s = \lfloor \bar{b}_h \rfloor, \quad s \ge 0.$$

If we subtract from the above equation the equation of the tableau

$$x_{\beta[h]} + \sum_{j \in N} \bar{a}_{hj} x_j = \bar{b}_h,$$

we obtain the equivalent equation

$$-\sum_{j \in N} \bar{a}_{hj} x_j + s = \lfloor \bar{b}_h \rfloor - \bar{b}_h.$$

Juxtaposing the above equation to the previous tableau, we obtain the problem

$$\begin{array}{lllrrll}
\max & z & & & & & \\
& -z & & + \sum_{j \in N} \bar{c}_j x_j & & = & -z_B & \\
& & x_{\beta[i]} & + \sum_{j \in N} \bar{a}_{ij} x_j & & = & \bar{b}_i, & i = 1, \ldots, m \\
& & & - \sum_{j \in N} \bar{a}_{hj} x_j & + \quad s & = & \lfloor \bar{b}_h \rfloor - \bar{b}_h & \\
& & & x, & s & \ge & 0. &
\end{array}$$

Note that the above problem is already in tableau form with respect to the dual feasible basis $B \cup \{s\}$. Furthermore, the basis is dual feasible because the reduced costs are all nonpositive (since $c_j \le 0$ for all $j \in N$). The right-hand-side of the new constraint (relative to the basic variable s) is $\lfloor \bar{b}_h \rfloor - \bar{b}_h < 0$, hence the basis is not primal feasible. We can therefore apply the dual simplex method to solve the new problem to optimality starting from the current dual feasible basis rather than restarting from scratch.

Note that the procedure for solving the LP relaxation of (P'') is similar. In this case we introduce a new nonnegative continuous surplus variable s for the new inequality $x_{\beta[h]} \ge \lceil \bar{b}_h \rceil$, to get the equation

$$x_{\beta[h]} - s = \lceil \bar{b}_h \rceil, \quad s \ge 0.$$

Subtracting the above to the equation of the tableau

$$x_{\beta[h]} + \sum_{j \in N} \bar{a}_{hj} x_j = \bar{b}_h,$$

we obtain the equivalent equation

$$\sum_{j \in N} \bar{a}_{hj} x_j + s = \bar{b}_h - \lceil \bar{b}_h \rceil.$$

Juxtaposing the above equation to the previous tableau, we obtain the problem

$$\begin{array}{lllrrll}
\max & z & & & & & \\
& -z & & + \sum_{j \in N} \bar{c}_j x_j & & = & -z_B & \\
& & x_{\beta[i]} & + \sum_{j \in N} \bar{a}_{ij} x_j & & = & \bar{b}_i, & i = 1, \ldots, m \\
& & & \sum_{j \in N} \bar{a}_{hj} x_j & + s & = & \bar{b}_h - \lceil \bar{b}_h \rceil & \\
& & & x, & s & \ge & 0. &
\end{array}$$

As before, the above problem is already in tableau form with respect to the dual feasible basis $B \cup \{s\}$. Since $\bar{b}_h - \lceil \bar{b}_h \rceil < 0$, the basis is not primal feasible. We can therefore apply the dual simplex method to solve the new problem.

Example. Consider the following integer program.

$$\begin{array}{rrrrcr}
\max & -5x_1 & -2x_2 & & & \\
& -x_1 & & -x_3 & \leq & -2 \\
& -x_1 & -x_2 & +4x_3 & \leq & -1 \\
& \multicolumn{3}{l}{x_1, x_2, x_3 \geq 0} & & \\
& \multicolumn{3}{l}{x_1, x_2, x_3 \in \mathbb{Z}} & &
\end{array}$$

We introduce slack variables x_4, x_5 to bring the problem in standard form, thus obtaining the mixed-integer programming problem

$$\begin{array}{lrrrrrcr}
\max z & & & & & & & \\
z & +5x_1 & +2x_2 & & & & = & 0 \\
& -x_1 & & -x_3 & +x_4 & & = & -2 \\
& -x_1 & -x_2 & +4x_3 & & +x_5 & = & -1 \\
& \multicolumn{3}{l}{x_1, \ldots, x_5 \geq 0} & & & & \\
& \multicolumn{3}{l}{x_1, x_2, x_3 \in \mathbb{Z}} & & & &
\end{array} \qquad (P_0)$$

We apply the simplex method to solve the linear programming relaxation (we have already solved this LP in week's III problem set, exercise 3), obtaining the optimal tableau:

$$\begin{array}{lrrrrrcr}
\max z & & & & & & & \\
z & & +x_2 & & +4x_4 & +x_5 & = & -9 \\
& & -\frac{1}{5}x_2 & +x_3 & -\frac{1}{5}x_4 & +\frac{1}{5}x_5 & = & \frac{1}{5} \\
& x_1 & +\frac{1}{5}x_2 & & -\frac{4}{5}x_4 & -\frac{1}{5}x_5 & = & \frac{9}{5} \\
& & \multicolumn{3}{l}{x_1, \ldots, x_5 \geq 0} & & &
\end{array}$$

We branch on x_1, thus creating subproblems $P_1 = (P_0) \cap \{x_1 \leq 1\}$ and $P_2 = (P_0) \cap \{x_1 \geq 2\}$.

To solve the LP relaxation of (P_1), we introduce a new (continuous) slack variable x_6 and write the tableau with respect to the basis $\{1, 3, 6\}$:

$$\begin{array}{lrrrrrrcr}
\max z & & & & & & & & \\
z & & +x_2 & & +4x_4 & +x_5 & & = & -9 \\
& & -\frac{1}{5}x_2 & +x_3 & -\frac{1}{5}x_4 & +\frac{1}{5}x_5 & & = & \frac{1}{5} \\
& x_1 & +\frac{1}{5}x_2 & & -\frac{4}{5}x_4 & -\frac{1}{5}x_5 & & = & \frac{9}{5} \\
& & -\frac{1}{5}x_2 & & +\frac{4}{5}x_4 & +\frac{1}{5}x_5 & +x_6 & = & -\frac{4}{5} \\
& & \multicolumn{3}{l}{x_1, \ldots, x_5 \geq 0} & & & &
\end{array}$$

At this point we will apply the dual simplex method to solve the LP relaxation of (P_1).

Similarly, to solve the LP relaxation of (P_2), we introduce a new (continuous) surplus variable x_6 and write the tableau with respect to the basis $\{1, 3, 6\}$:

$$\begin{array}{lrrrrrrcr}
\max z & & & & & & & & \\
z & & +x_2 & & +4x_4 & +x_5 & & = & -9 \\
& & -\frac{1}{5}x_2 & +x_3 & -\frac{1}{5}x_4 & +\frac{1}{5}x_5 & & = & \frac{1}{5} \\
& x_1 & +\frac{1}{5}x_2 & & -\frac{4}{5}x_4 & -\frac{1}{5}x_5 & & = & \frac{9}{5} \\
& & +\frac{1}{5}x_2 & & -\frac{4}{5}x_4 & -\frac{1}{5}x_5 & +x_6 & = & -\frac{1}{5} \\
& & \multicolumn{3}{l}{x_1, \ldots, x_5 \geq 0} & & & &
\end{array}$$

At this point we will apply the dual simplex method to solve the LP relaxation of (P_2).

Chapter 2

Good formulations

We consider a general mixed-integer programming problem

$$\begin{aligned} z_I = \max c^\top x & \\ Ax \le b & \\ x \ge 0 & \\ x_i \in \mathbb{Z}, & \qquad i \in I. \end{aligned} \tag{2.1}$$

where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ and $I \subseteq \{1, \dots, n\}$. As usual, let

$$X = \{x \in \mathbb{R}^n \mid Ax \le b,\, x \ge 0,\, x_i \in \mathbb{Z} \text{ for all } i \in I\}$$

be the *feasible region* of the problem.

We have defined the linear relaxation of (2.1) to be the LP problem

$$\begin{aligned} z_L = \max c^\top x \\ Ax \le b \\ x \ge 0 \end{aligned} \tag{2.2}$$

Note that the linear relaxation is not unique (see Figure 2.1). Indeed, given a matrix $A' \in \mathbb{R}^{m' \times n}$ and a vector $b' \in \mathbb{R}^{m'}$, we say that the system of linear inequalities

$$\begin{aligned} A'x \le b' \\ x \ge 0 \end{aligned}$$

is a *formulation* for the set X above if

$$X = \{x \in \mathbb{R}^n \mid A'x \le b',\, x \ge 0,\, x_i \in \mathbb{Z} \text{ for all } i \in I\}.$$

In this case, the LP problem

$$\begin{aligned} z'_L = \max c^\top x \\ A'x \le b' \\ x \ge 0 \end{aligned} \tag{2.3}$$

is also a linear relaxation for (2.1). Obviously the optimal value z'^L of the latter relaxation might differ from the optimal value z^L of the linear relaxation (2.2).

Is is clear that X may have infinitely many formulations, and so a MILP problem (2.1) can have infinitely many different linear relaxations. Which are the best relaxations?

Given two formulations for X,

$$\begin{aligned} Ax \le b \\ x \ge 0 \end{aligned} \qquad \text{and} \qquad \begin{aligned} A'x \le b \\ x \ge 0 \end{aligned}$$

we say that the first formulation is *better* than the second if the polyhedron determined by the first system of constraints is contained in the one determined by the second system, that is, if

$$\{x \in \mathbb{R}^n \mid Ax \le b,\, x \ge 0\} \subseteq \{x \in \mathbb{R}^n \mid A'x \le b',\, x \ge 0\}.$$

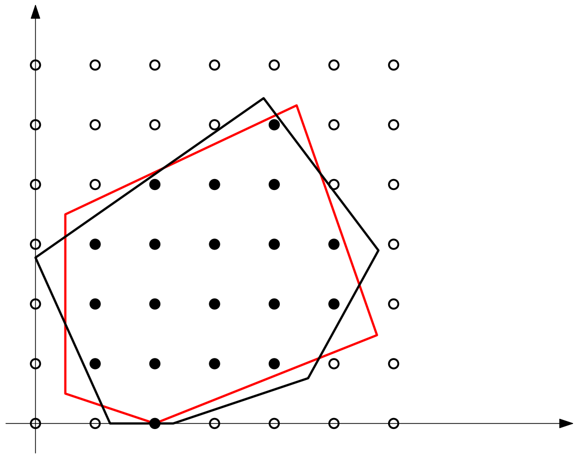


Figure 2.1: Two formulations for the same feasible set X (represented by the black dots).

The above notion is justified by the fact that, if $Ax \leq b$, $x \geq 0$ is a better formulation than $A'x \leq b'$, $x \geq 0$, then

$$z_I \leq z_L \leq z'_L,$$

and thus the linear relaxation $\{\max c^\top x \mid Ax \leq b,\ x \geq 0\}$ gives a tighter lower-bound on the MILP optimal value z_I than the one given by the linear relaxation $\{\max c^\top x \mid A'x \leq b',\ x \geq 0\}$. As we have seen, tighter lower-bounds typically allows us to consider less sub-problems in the branch-and-bound procedure, since more nodes could be pruned.

2.1 Examples of formulations

2.1.1 The knapsack problem

We are given a knapsack that can carry a maximum weight b and n types of items, where an item of type i has weight $a_i > 0$. We want to load the knapsack with items (possibly several items of the same type) without exceeding the knapsack capacity b. To model this, we associate a variable x_i that represents the number of items of type i to be loaded. Then the following knapsack set:

$$S := \{x \geq 0 \text{ integral} : \sum_{i=1}^{n} a_i x_i \leq b\}$$

represents all the feasible loads.
If an item of type i has value c_i, the problem of loading the knapsack so as to maximize the total value of the load can be modeled as follows:

$$\max\{\sum_{i=1}^{n} c_i x_i : x \in S\}.$$

If only one unit of each item type can be selected, we use binary variables instead of general integers. This is the $0,1$ knapsack set:

$$K := \{x \in \{0,1\}^n : \sum_{i=1}^{n} a_i x_i \leq b\}.$$

2.1.2 Comparing formulations

Given a $0,1$ knapsack set $K := \{x \in \{0,1\}^n : \sum_{i=1}^n a_i x_i \leq b\}$, a subset C of indices is a *minimal cover* if $\sum_{i \in C} a_i > b$ and $\sum_{i \in C \setminus \{j\}} a_i \leq b$ for every $j \in C$. That is, the knapsack cannot contain all items in

C, but every proper subset of C can be loaded. Consider the set

$$K^C := \{x \in \{0,1\}^n : \sum_{i \in C} x_i \leq |C| - 1 \text{ for every minimal cover } C \text{ of } S\}.$$

Lemma 2.1. *The sets K and K^C coincide.*

Proof. If suffices to show that (i) if C is a minimal cover of K, the inequality $\sum_{i \in C} x_i \leq |C| - 1$ is valid for K and (ii) the inequality $\sum_{i=1}^n a_i x_i \leq b$ is valid for K^C. The first statement follows from the fact that the knapsack cannot contain all the items in a minimal cover.
Let $\bar{x}$ be a vector in K^C and let $J := \{j : \bar{x}_j = 1\}$. Suppose $\sum_{i=1}^n a_i \bar{x}_i > b$ or equivalently $\sum_{i \in J} a_i > b$. Let C be a minimal subset of J such that $\sum_{i \in C} a_i > b$. Then obviously C is a minimal cover and $\sum_{i \in C} \bar{x}_i = |C|$. This contradicts the assumption $\bar{x} \in K^C$ and the second statement is proved. $\square$

So K and K^C describe the same set but the constraints that define them look quite different. K is defined by a single inequality with nonnegative integer coefficients. S^C is defined by many inequalities (their number may be exponential in n) whose coefficients are $0, 1$. However, adding all the covers inequalities can yield a much stricter formulation.

We conclude this section with an example of a $0,1$ knapsack set where the minimal cover formulation is better than the knapsack formulation. Consider the following knapsack set

$$K := \{x \in \{0,1\}^4 : \ 6x_1 + 6x_2 + 6x_3 + 5x_4 \leq 16\}.$$

Its minimal cover formulation is

$$\begin{array}{rlllll} K^C := \{x \in \{0,1\}^4 : & x_1 & +x_2 & +x_3 & & \leq 2 \\ & x_1 & +x_2 & & +x_4 & \leq 2 \\ & x_1 & & +x_3 & +x_4 & \leq 2 \\ & & x_2 & +x_3 & +x_4 & \leq 2\}. \end{array}$$

The corresponding linear relaxations are the polyhedra

$$P := \{x \in [0,1]^4 : \ 6x_1 + 6x_2 + 6x_3 + 5x_4 \leq 16\}, \text{ and}$$

$$\begin{array}{rlllll} P^C := \{x \in [0,1]^4 : & x_1 & +x_2 & +x_3 & & \leq 2 \\ & x_1 & +x_2 & & +x_4 & \leq 2 \\ & x_1 & & +x_3 & +x_4 & \leq 2 \\ & & x_2 & +x_3 & +x_4 & \leq 2\}. \end{array}$$

respectively. By adding up the four inequalities in P^C we get

$$3x_1 + 3x_2 + 3x_3 + 3x_4 \leq 8$$

which, together with $x_4 \geq 0$, implies $6x_1 + 6x_2 + 6x_3 + 5x_4 \leq 16$. Thus $P^C \subseteq P$. The inclusion is strict since, for instance $(\frac{5}{6}, 1, 0, 1) \in P \setminus P^C$. In other words, the minimal cover formulation is strictly better than the knapsack formulation in this case. One can also construct examples where the knapsack formulation is strictly better than the minimal cover formulation.

2.1.3 The traveling salesman problem

A traveling salesman must visit n cities and return to the city he started from. We will call this a *tour*. Given the cost c_{ij} of traveling from city i to city j, for each $1 \leq i, j \leq n$ with $i \neq j$, in which order should the salesman visit the cities to minimize the total cost of his tour? This problem is the famous *traveling salesman problem* (TSP). We give three different formulations.

To model the problem as an integer program, we introduce a binary variable x_{ij} for all $i, j = 1, \dots, n$ such that $i \neq j$. The variable x_{ij} takes value 1 if the tour visits city j immediately after city i, and 0 otherwise. Given a set of cities $S \subset \{1, \ldots, n\}$, let $\bar{S}$ denote its complement, namely $\bar{S} := \{1, \ldots, n\} \backslash S$. The TSP problem can be formulated as follows.

$$\min \quad \sum_i \sum_{j \neq i} c_{ij} x_{ij} \tag{2.4}$$

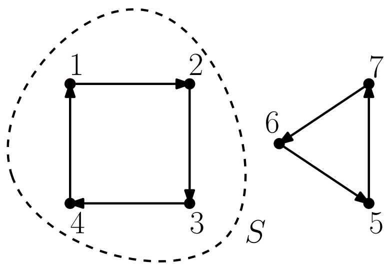
$$\sum_{j \neq i} x_{ij} = 1 \qquad \text{for } i = 1, \ldots, n \tag{2.5}$$

$$\sum_{i \neq j} x_{ij} = 1 \qquad \text{for } j = 1, \ldots, n \tag{2.6}$$

$$\sum_{i \in S, j \in \bar{S}} x_{ij} \geq 1 \qquad \text{for } \emptyset \subsetneqq S \subsetneqq \{1, \ldots, n\},\ |S|, |\bar{S}| \geq 2 \tag{2.7}$$

$$x_{ij} = 0 \text{ or } 1 \quad \text{for } i, j = 1, \ldots, n, \quad i \neq j. \tag{2.8}$$

Constraints (2.5)-(2.6) guarantee that the traveling salesman visits each city exactly once. Note however that constraints (2.5)-(2.6) alone are not sufficient, since we might have situations as depicted in the following figure which corresponds to the solution $x_{12} = x_{23} = x_{34} = x_{41}$, $x_{56} = x_{67} = x_{75}$, and



$x_{ij} = 0$ otherwise. This solution satisfies (2.5)-(2.6), however it does not correspond to one tour, but to multiple subtours.

To prevent this from happening, we impose constraints (2.7), which are called the *subtour elimination constraints*. Indeed, given any proper subset S of the cities, $\emptyset \neq S \subset \{1, \ldots, n\}$, to "reach" the cities in $\bar{S}$ from the cities in S we need to cross S, and thus there must be a city in $\bar{S}$ that is preceded by a city in S. This condition is enforced by constraints (2.7). For example, in the situation in the above figure, if we choose $S = \{1, 2, 3, 4\}$, there is no arc that crosses S, hence all variables of the form x_{ij} with $i \in S$, $j \notin S$ have value 0. Thus such solution violates the subtour elimination constraints. Note that we only need the subtour elimination constraints only for sets S such that both S and $\bar{S}$ contain at least two elements; indeed, if $|S| = 1$ then the subtour elimination constraint relative to S is implied by the equation (2.5) relative to the only node in S, and if $|\bar{S}| = 1$ then the subtour elimination constraint relative to S is implied by the equation (2.6) relative to the only node in $\bar{S}$.

Notice that such formulation has an exponential number of constraints, since the number of proper subsets of $\{1, \ldots, n\}$ is 2^{n-1}. Despite the exponential number of constraints, this is the formulation that is most widely used in practice. Initially, one solves the LP relaxation that only has contains (2.5)-(2.6) and $0 \leq x_{ij} \leq 1$. The subtour elimination constraints are added later, on the fly, only when needed.

2.1.4 A smaller formulation

Miller, Tucker, and Zemlin found a way to avoid the subtour elimination constraints (2.7). They introduce an extra variables u_i for every city $i = 2, \ldots, n$, where u_i the position of city $i \geq 2$ in the tour, assuming that city 1 has position 1. Their formulation is identical to (2.4)-(2.8) except that the subtour eliminations (2.7) are replaced by the constraints

$$u_i - u_j + 1 \leq n(1 - x_{ij}) \qquad \text{for all } i \neq 1,\ j \neq 1. \tag{2.9}$$

Note that the above inequality forces u_j to be at least $u_i + 1$ whenever $x_{ij} = 1$, while if $x_{ij} = 0$ the constraint simply imposes that $u_i - u_j \leq n - 1$, which must be satisfied by any pair of cities i, j, since their position in the tour cannot differ by more than $n - 1$.

It is not difficult to verify that the Miller-Tucker-Zemlin (MTZ) formulation is correct. Indeed, if $\bar{x}$ is a $0,1$ vector representing a tour, then if we define u_i to be the position of city i in the tour, for $i \geq 2$, the constraints (2.9) are satisfied.

Conversely, if $\bar{x}$ is a $0,1$ vector that does not represent a tour, then (2.5)-(2.6) imply that there is at least one subtour that does not contain vertex 1. Suppose that the subtour has length k, and that it traverses the nodes $i_1, i_2, \ldots, i_k, i_1$. Summing up all the inequalities (2.9) around this subtour yields

$$\sum_{h=1}^{k-1}(u_{i_h} - u_{i_{h+1}} + 1) + (u_{i_k} - u_{i_1} + 1) \leq \sum_{h=1}^{k-1} n(1 - \bar{x}_{i_h i_{h+1}}) + n(1 - \bar{x}_{i_k i_1}).$$

Since the left-hand-side vanishes because $\bar{x}_{i_h i_{h+1}} = 1$ for all $h = 1, \ldots, k-1$ and $\bar{x}_{i_k i_1} = 1$, and since all the variables in the left-hand side cancel out since the appear exactly once with a $+1$ coefficient and once with a -1 coefficient, it follows that the above reduces to the inequality $k \leq 0$, which is inconsistent.

Therefore, if (2.5)-(2.6), (2.8), (2.9) are satisfied, $\bar{x}$ must represent a tour. Although the MTZ formulation is correct, it can be proven that it is worse than the subtour elimination formulation, and in practice it is observed that indeed it produces far weaker bounds.

This shows a counterintuitive fact: a "smaller" formulation (in the sense that the number of variables plus constraints is smaller) is not necessarily more effective. Indeed, even though solving the linear relaxation of the smaller formulation might be much faster, the bounds provided by such linear relaxation might be so weak that it might be worthwhile spending more time solving a linear relaxation with many more constraints.

2.1.5 Facility location

We need to decide where to open facilities at n possible locations in order to serve m customer. Let d_i be the annual demand of customer i, $i = 1, \ldots, m$, let c_{ij} be the unit cost of servicing client i from location j, and let f_j be the fixed cost of opening a facility at location j.

The facility location problem consists in deciding in which locations to open a facility, and from which facilities to service each client in order to satisfy demand, while minimizing the total cost.

To model the problem as a mixed-integer linear program, we introduce the following decision variables:

$$\begin{array}{rcl} y_{ij} & = & \text{fraction of annual demand provided from } j \text{ to } i; \\ x_j & = & \begin{cases} 1 & \text{if facility is built at } j; \\ 0 & \text{otherwise.} \end{cases} \end{array}$$

The objective function is

$$\min \quad \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} d_i y_{ij} + \sum_{j=1}^{n} f_j x_j.$$

We propose two possible formulations.

Aggregate formulation

$$\begin{array}{rcll} \sum_{j=1}^{n} y_{ij} & = & 1 & i = 1, \ldots, m \\ \sum_{i=1}^{m} y_{ij} & \leq & m x_j & j = 1, \ldots, n \\ y & \geq & 0 & \\ x & \in & \{0,1\}^n & . \end{array}$$

Disaggregate formulation

$$\begin{array}{rcll} \sum_{j=1}^{n} y_{ij} & = & 1 & i = 1, \ldots, m \\ y_{ij} & \leq & x_j & i = 1, \ldots, m,\ j = 1, \ldots, n \\ y & \geq & 0 & \\ x & \in & \{0,1\}^n & . \end{array}$$

Both formulations are correct, however the disaggregate formulation is better than the aggregate one. Indeed, let

$$P_1 = \left\{ (x,y) \in \mathbb{R}^n \times \mathbb{R}^m \,|\, \begin{array}{rcll} \sum_{j=1}^n y_{ij} & = & 1 & i = 1,\dots,m \\ \sum_{i=1}^m y_{ij} & \leq & m x_j & j = 1,\dots,n \\ y & \geq & 0 & \\ 0 \leq x_j & \leq & 1 & j = 1,\dots,n \end{array} \right\}$$

$$P_2 = \left\{ (x,y) \in \mathbb{R}^n \times \mathbb{R}^m \,|\, \begin{array}{rcll} \sum_{j=1}^n y_{ij} & = & 1 & i = 1,\dots,m \\ y_{ij} & \leq & x_j & i = 1,\dots,m,\ j = 1,\dots,n \\ y & \geq & 0 & \\ 0 \leq x_j & \leq & 1 & j = 1,\dots,n \end{array} \right\}$$

We have that $P_2 \subsetneq P_1$. To see that $P_2 \subseteq P_1$, we show that every point in P_2 satisfies all the inequalities defining P_1. Note that the only inequalities where the two formulations differ are the inequalities

$$\sum_i y_{ij} \leq m x_j \tag{2.10}$$

for all possible location $j = 1,\dots,n$. Any point in P_2 satisfies

$$y_{ij} \leq x_j \tag{2.11}$$

for all possible clients $i = 1,\dots,m$ and all possible facilities $j = 1,\dots,n$. If, for a given location j we take the sum of the inequalities (2.11) for $i = 1,\dots,m$, we obtain precisely inequality (2.10). This shows that every point in P_2 satisfies inequalities (2.10), so every point in P_2 is also in P_1.

To see that the containment is strict, we only need to give an example of a point in P_1 that is not in P_2. Consider therefore the case where $n = 2$ and $m = 4$, and consider the point $(\bar{x}, \bar{y})$ defined by $\bar{x}_1 = \bar{x}_2 = 1/2$, $\bar{y}_{11} = \bar{y}_{21} = \bar{y}_{32} = \bar{y}_{42} = 1$, $\bar{y}_{31} = \bar{y}_{41} = \bar{y}_{12} = \bar{y}_{22} = 0$. One can see that $(\bar{x}, \bar{y})$ satisfies the inequalities of the aggregate formulation (note that the inequalities (2.10) become $y_{1j} + y_{2j} + y_{3j} + y_{4j} \leq 4x_j$ for $j = 1, 2$), but $(\bar{x}, \bar{y})$ does not satisfy the inequalities of the disaggregate formulation, since the inequality $y_{11} \leq x_1$ is violated because $\bar{y}_{11} = 1 \not\leq 1/2 = \bar{x}_1$.

2.2 Ideal formulations

From the previous discussion, it appears that it is sensible to consider the "best possible formulation" for a set X, meaning the one for which the linear relaxation has the smallest feasible region (with respect to inclusion). Here we formalize this concept.

As we have seen, a formulation for the set X is a system of inequalities defining a polyhedron P such that $X \subseteq P$ and

$$X = \{x \in P \,|\, x_i \in \mathbb{Z}\ \forall i \in I\}.$$

Hence we will also say that the polyhedron P is a formulation for X. Given two polyhedra P and P' that are formulations for X, we said that the formulation P is *better* than P' if $P \subset P'$. Recall that any polyhedron is a convex set.

Given any set $S \subseteq \mathbb{R}^n$, the *convex hull of* S is the (unique) minimal convex set containing S. We denote such set by $\mathrm{conv}(S)$. In other words, $\mathrm{conv}(S)$ is the convex subset of $\mathbb{R}^n$ such that $S \subseteq \mathrm{conv}(S)$ and $\mathrm{conv}(S) \subseteq C$ for every convex set C containing X.

Hence, given any formulation $P = \{x \,|\, Cx \leq d,\ x \geq 0\}$ for X, then, since P is a convex set containing X, we have that

$$X \subseteq \mathrm{conv}(X) \subseteq P.$$

Therefore $\mathrm{conv}(X)$ must be contained in the feasible region of the linear relaxation of every formulation for X. In particular, if $\mathrm{conv}(X)$ is itself a polyhedron, then it follows that the system of linear inequalities defining $\mathrm{conv}(X)$ is the best possible formulation.

The following theorem shows that, indeed, $\mathrm{conv}(X)$ is a polyhedron (assuming that the data of our MILP problem are all rational, which for practical purposes can always be assumed).

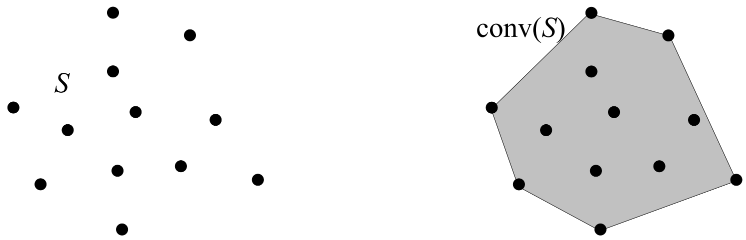


Figure 2.2: A set and its convex hull.

Theorem 2.2. (Fundamental theorem of Integer Programming) *Given* $A \in \mathbb{Q}^{m\times n}$ *and* $b \in \mathbb{Q}^m$, *let* $X = \{x \in \mathbb{R}^n \,|\, Ax \le b,\ x \ge 0,\ x_i \in \mathbb{Z} \text{ for all } i \in I\}$. *Then* $\operatorname{conv}(X)$ *is a polyhedron.*
That is, there exists a matrix $\tilde{A} \in \mathbb{Q}^{\tilde{m}\times n}$ *and a vector* $\tilde{b} \in \mathbb{Q}^{\tilde{m}}$ *such that*

$$\operatorname{conv}(X) = \{x \in \mathbb{R}^n \,|\, \tilde{A}x \le \tilde{b},\ x \ge 0\}.$$

In light of the above theorem, given $\tilde{A} \in \mathbb{Q}^{\tilde{m}\times n}$ and $\tilde{b} \in \mathbb{Q}^{\tilde{m}}$ such that $\operatorname{conv}(X) = \{x \in \mathbb{R}^n \,|\, \tilde{A}x \le \tilde{b},\ x \ge 0\}$, we will say that $\tilde{A}x \le \tilde{b},\ x \ge 0$ is the *ideal formulation* for X.

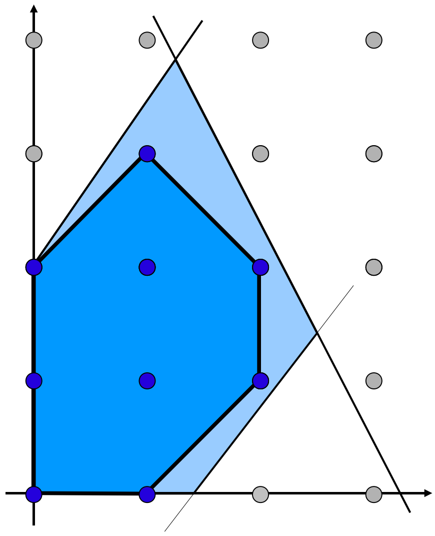


Figure 2.3: A formulation for a set of integer points, and the ideal formulation.

The following theorem further justifies the term "ideal formulation".

Theorem 2.3. *Given a matrix* $A \in \mathbb{Q}^{m\times n}$ *and a vector* $b \in \mathbb{Q}^m$, *let* $X = \{x \in \mathbb{R}^n \,|\, Ax \le b,\ x \ge 0,\ x_i \in \mathbb{Z} \text{ for all } i \in I\}$. *For any* $c \in \mathbb{R}^n$, *let*

$$z_I = \max_{x \in X} c^\top x \quad \text{and} \quad \tilde{z} = \max_{x \in \operatorname{conv}(X)} c^\top x .$$

Then $z_I = \tilde{z}$. *Furthermore, all vertices of the polyhedron* $\operatorname{conv}(X)$ *are elements of* X.

Proof. We first show that $z_I = \tilde{z}$. The direction "$z_I \le \tilde{z}$" is obvious, since $X \subseteq \operatorname{conv}(X)$. For the direction"$\tilde{z} \le z_I$", consider the half-space $H = \{x \in \mathbb{R}^n \,|\, c^\top x \le z_I\}$. Since z_I is the optimal value of the problem $\max\{c^\top x \,|\, x \in X\}$, it follows that $X \subseteq H$. But H is a convex set containing X, therefore $\operatorname{conv}(X) \subseteq H$. It follows then that $c^\top x \le z_I$ for all $x \in \operatorname{conv}(X)$. Thus $\max\{c^\top x \,|\, x \in \operatorname{conv}(X)\} \le z_I$.

By Theorem 2.2, $\operatorname{conv}(X)$ is a polyhedron. Consider a vertex $\bar{x}$ of $\operatorname{conv}(X)$. By definition of vertex, there exists some vector $c \in \mathbb{R}^n$ such that $c \ne 0$ and $c^\top x < c^\top \bar{x}$ for all $x \in \operatorname{conv}(X) \setminus \{x\}$. Thus $\bar{x}$ is the unique optimal solution of $\max\{c^\top x \mid x \in \operatorname{conv}(X)\}$. By the previous point, $\max\{c^\top x \mid x \in X\} = \max\{c^\top x \mid x \in \operatorname{conv}(X)\}$, but since x is the unique optimal solution of the problem on the right, and since $X \subseteq \operatorname{conv}(X)$, it follows that $\bar{x}$ is an optimal solution to $\max\{c^\top x \mid x \in X\}$, so in particular $\bar{x}$ belongs to X. □

Let $\tilde{A}x \leq \tilde{b}$ be the ideal formulation for the set X as defined as in Theorem 2.3. By Theorem 2.3, any basic optimal solution of the LP problem

$$\begin{aligned} \tilde{z} = \max c^\top x \\ \tilde{A}x \leq \tilde{b} \\ x \geq 0. \end{aligned} \tag{2.12}$$

is also an optimal solution of the original MILP problem

$$\begin{aligned} z_I = \max c^\top x & \\ Ax \leq b & \\ x \geq 0 & \\ x_i \in \mathbb{Z}, & \qquad i \in I. \end{aligned} \tag{2.13}$$

This shows that solving the MILP problem (2.13) is equivalent to solving the LP problem (2.12). Thus, in principle, solving a mixed-integer linear programming problem is equivalent to solving a linear programming problem. There are, however, to major hurdles that make using the ideal formulation often impossible.

- The ideal formulation of a MILP problem is usually not known, and computing it can be extremely difficult (and, in fact, computationally intractable most of the times),

- The ideal formulation could have an exponential number of constraints compared to the original system $Ax \leq b$, and so linear programming algorithms could be quite ineffective on the ideal formulation because of its size.

Chapter 3

Assignment Problem and Total unimodularity

3.1 The assignment problem

A *graph* $G = (V, E)$ consists of a finite set V of elements, called *nodes* and a set E of unordered pairs of nodes, called *edges*. Graphs are usually depicted by points, representing the vertices, and lines joining pairs of points, representing the edges.

A *matching* in G is a set of edges $M \subseteq E$ such that the elements of M are pairwise disjoint. In other words, $M \subseteq E$ is a matching of G if every node of G is contained in at most one edge of M.

The *maximum cardinality matching problem* is the following: given a graph $G = (V, E)$, determine a matching M with the largest possible number of edges.

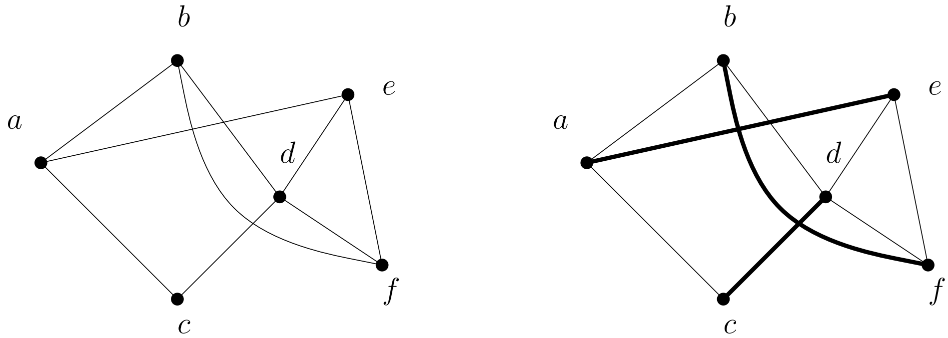


Figure 3.1: The graph $G = (V, E)$ where $V = \{a, b, c, d, e, f\}$ and $E = \{ab, ac, ae, bd, be, cd, de, df, ef\}$. The boldfaced edges on the graph on the right represent the elements of the matching $M = \{ae, bf, cd\}$.

We can formulate the maximum cardinality matching problem as an integer programming problem as follows. Assign a binary variable x_e to each edge $e \in E$, where

$$x_e = \begin{cases} 1 & \text{if } e \text{ is in the matching,} \\ 0 & \text{otherwise,} \end{cases} \qquad e \in E.$$

If $M = \{e \in \ | x_e = 1\}$ is a matching, the cardinality of M is given by

$$\sum_{e \in E} x_e.$$

To make sure that the variables x_e, $e \in E$, define a matching, we need to guarantee that each node $v \in V$ is contained in at most one edge $e \in E$ such that $x_e = 1$. We express this through the linear constraint

$$\sum_{u \in V : uv \in E} x_{uv} \leq 1, \quad v \in V.$$

Hence the maximum cardinality matching problem can be formulated as

$$\begin{aligned} \max \sum_{e \in E} x_e & \\ \sum_{uv \in E} x_{uv} \leq 1, & \qquad v \in V \\ x_e \geq 0, & \qquad e \in E \\ x_e \in \mathbb{Z}, & \qquad e \in E. \end{aligned} \tag{3.1}$$

The above is not the ideal formulation. For example, suppose G is just a triangle, with vertices $\{1,2,3\}$ and edges $\{12,13,23\}$. The system defined by the constraints in (3.1) is

$$\begin{array}{cccccc} x_{12} & +x_{13} & & \leq & 1 \\ x_{12} & & +x_{23} & \leq & 1 \\ & x_{13} & +x_{23} & \leq & 1. \\ & x \geq 0 & & & \end{array}$$

Clearly the maximum cardinality matching in a triangle consists of just one edge, because any two edges have an endnode in common. It follows that the above system of constraints does not have any feasible integer solution. However, the point $(1/2, 1/2, 1/2)$ is feasible and it has value $3/2$ in the objective function. Since the optimal integer solution has value 1 while optimal solution of the linear relaxation has value $3/2 > 1$, it follows that the above formulation is not the ideal one.

It turns out that the above type of formulation is ideal for an important special class of graphs. The graph G is said to be *bipartite* if its node set V can be partitioned into two disjoint sets V_1, V_2 such that every edge $uv \in V$, exactly one node among u and v is in V_1, and the other is in V_2. The pair of sets V_1, V_2 are said a *bipartition* of G.

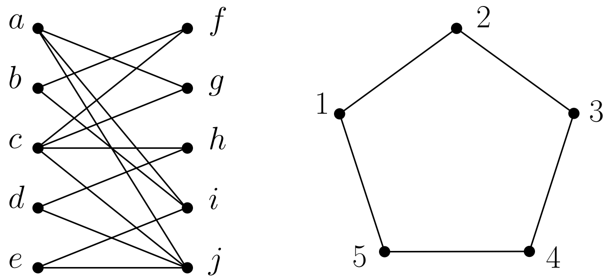


Figure 3.2: The graph on the left is bipartite, with bipartition V_1, V_2 where $V_1 = \{a,b,c,d,e\}$ and $V_2 = \{f,g,h,i,j\}$. The graph on the right is not bipartite; indeed, if a bipartition V_1, V_2 existed, say with node 1 in V_1, then 2 should be in V_2 because $\{1,2\}$ is an edge. Similarly, 3 should be in V_1 and 4 in V_2. Now node 5 is adjacent to $1 \in V_1$ and $4 \in V_2$, so it cannot be placed in either side.

Assignment problem: *Given a bipartite graph $G = (V,E)$ with bipartition V_1, V_2 such that $|V_1| = |V_2|$ and costs c_e on every edge $e \in E$, one wants to assign to every node u in V_1 exactly one node v in V_2 so that $uv \in E$, every element of V_2 is assigned to exactly one element of V_1, and the total cost of the pairs selected is minimized.*

Note that an assignment is a *perfect matching*, that is, a matching that covers all nodes of the graph. To formulate the assignment problem as an integer programming problem, as usual we have a binary variable x_{uv} for every $u \in V_1$, $v \in V_2$ such that $uv \in E$, where

$$x_{uv} = \begin{cases} 1 & \text{if } u \text{ is assigned to } v, \\ 0 & \text{otherwise.} \end{cases}$$

The total cost determined by x is therefore

$$\sum_{uv \in E} c_{uv} x_{uv}.$$

The condition that every node u in V_1 is assigned to exactly one adjacent node $v \in V_2$ is expressed by the constraints

$$\sum_{uv \in E} x_{uv} = 1, \quad u \in V_1,$$

while the fact that every $v \in V_2$ is assigned to exactly one adjacent node $u \in V_1$ is expressed by the constraints

$$\sum_{uv \in E} x_{uv} = 1, \quad v \in V_2.$$

The assignment problem can therefore be written as

$$\begin{array}{rcll} \min \sum_{uv \in E} c_{uv} x_{uv} & & & \\ \sum_{v: uv \in E} x_{uv} & = & 1, & u \in V_1, \\ \sum_{u: uv \in E} x_{uv} & = & 1, & v \in V_2, \\ x_{uv} & \geq & 0, & uv \in E \end{array} \tag{3.2}$$

Note that we do not need to include the constraints $x_{uv} \leq 1$ for every $uv \in E$, as they are all implied by the other constraints.

Theorem 3.1. *Formulation* (3.2) *is ideal.*

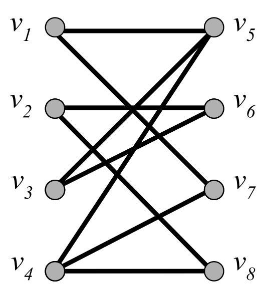
To prove Theorem (3.1), let us write the above problem in matrix form.

Definition 3.2. *Given a graph $G = (V, E)$, the* incidence matrix of G *is the* $0, 1$ *matrix $A(G)$ with $|V|$ rows and $|E|$ columns, where each row is indexed by a node of V and each column is indexed by an edge in E, and where the entry $a_{v,e}$ of $A(G)$ in the row indexed by node v and in the column indexed by edge e is defined by*

$$a_{v,e} = \begin{cases} 1 & \text{if } v \in e, \\ 0 & \text{otherwise.} \end{cases}$$

Note that every column of $A(G)$, relative to some edge $uv \in E$, has exactly 2 entries of value 1 (namely the ones corresponding to u and v), and all the others of value 0.

Furthermore, if G is bipartite, then every column of $A(G)$ has exactly one entry of value 1 in the rows of $A(G)$ indexed by nodes in V_1, and exactly one entry of value 1 in the rows of $A(G)$ indexed by nodes in V_2. The next pictures shows a bipartite graph and the corresponding incidence matrix.



	v_1v_5	v_1v_7	v_2v_6	v_2v_8	v_3v_5	v_3v_6	v_4v_5	v_4v_7	v_4v_8
v_1	1	1	0	0	0	0	0	0	0
v_2	0	0	1	1	0	0	0	0	0
v_3	0	0	0	0	1	1	0	0	0
v_4	0	0	0	0	0	0	1	1	1
v_5	1	0	0	0	1	0	1	0	0
v_6	0	0	1	0	0	1	0	0	0
v_7	0	1	0	0	0	0	0	1	0
v_8	0	0	0	1	0	0	0	0	1

Note that, given a row of $A(G)$ relative to a node $u \in V$, the entries of value 1 in the row are the ones indexed by edges containing u.

Thus the formulation of the assignment problem given in (3.2) can be written in the matrix form as

$$\begin{gathered} \min c^T x \\ A(G)\, x = \mathbf{1} \\ x \geq 0 \end{gathered}$$

where by $\mathbf{1}$ we denote the vector with all components equal to 1.

To prove that the above is the ideal formulation, we will show that all the basic solutions of the system $A(G)\ x = \mathbf{1}$, $x \geq 0$ are integer. This will follows from a very useful general property: total unimodularity

3.2 Totally unimodular matrices

Definition 3.3. *A matrix A is said* totally unimodular *if, for every square submatrix B of A,* $\det(B) \in \{0, +1, -1\}$.

Note that, since every entry of A is a square 1×1 matrix, every totally unimodular matrix has entries with value 0, +1 or −1.

Recall the following simple linear algebra fact, known as Cramer's rule. Given a square matrix $B \in \mathbb{R}^{m \times m}$, and given indices $i, j \in \{1, \dots, m\}$, let B^{ij} be the matrix obtained from a B by removing the jth row and the ith column. If B is non-singular, then the entry in position (i, j) of the inverse B^{-1} of B is

$$(B^{-1})_{i,j} = (-1)^{i+j} \frac{\det(B^{ij})}{\det(B)}.$$

In particular, the above implies that, if B is integer, then all the entries of B^{-1} are integer numbers divided by the determinant of B. Thus, if B is integer and $\det(B) = \pm 1$, then B^{-1} is an integer matrix. This proves the following.

Observation 3.4. *If A is a totally unimodular matrix, then B^{-1} is integer for every square nonsingular submatrix B of A.*

Theorem 3.5. *Let $A \in \mathbb{R}^{m \times n}$ be totally unimodular matrix, and let $b \in \mathbb{Z}^m$. The all basic solutions of*

$$\begin{aligned} Ax &= b \\ x &\geq 0 \end{aligned}$$

are integer.

Proof: We recall that, given a basis B of A, the basic solution $\bar{x}$ of $Ax = b$, $x \geq 0$ associated with B is

$$\begin{aligned} \bar{x}_B &= {A_B}^{-1} b, \\ \bar{x}_N &= 0. \end{aligned}$$

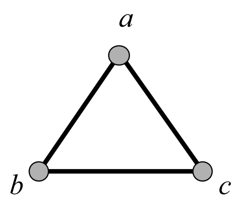
Since A is totally unimodular, then by Observation 3.4 A_B^{-1} is an integer matrix. Since b is an integer vector, then $A_B^{-1} b$ is and integer vector. Thus $\bar{x} \in \mathbb{Z}^n$. □

Theorem 3.6. *The incidence matrix of a bipartite graph is totally unimodular.*

Proof: Let $G = (V, E)$ be a bipartite graph with bipartition V_1, V_2. We need to show that every square submatrix of $A(G)$ has determinant in $\{0, 1, -1\}$. Suppose by contradiction this is not the case. Among all square submatrices of $A(G)$ with determinant different from 0, 1, and −1, let B be the one with the smallest possible number of rows. There are three cases.
a) B has a column of all zeros. In this case $\det(B) = 0$, a contradiction.
b) B has a column with exactly one entry with value 1. Assume that the jth column of B has value 1 in the ith row, and value 0 in all other entries. For the well known Laplace rule to compute determinants, if we denote by B' the matrix obtained form B by removing row i and column j, we have $\det(B) = (-1)^{i+j} \det(B')$. Since B' is square and has less rows than B, then $\det(B') \in \{0, +1, -1\}$, hence $\det(B) = (-1)^{i+j} \det(B') \in \{0, +1, -1\}$, a contradiction.
c) Every column of B has exactly two entries with value 1. Every column of B has one entry with value 1 in the rows indexed by V_1 and one entry with value 1 in the rows indexed by V_2. Thus the sum of all rows of B indexed by V_1 minus the sum of all rows of B indexed by V_2 is the 0 vector. This shows that the rows of B are linearly independent, thus $\det(B) = 0$, a contradiction. □

Together, Theorem 3.5 and Theorem 3.6 imply that all basic solutions of (3.2) are integer. This proves Theorem 3.1.

Note that the hypothesis in Theorem 3.6 that the graph is bipartite is crucial. As we have seen, the formulation of the matching problem is not ideal for non-bipartite graphs. Let us again consider the example of a triangle.



$$\begin{array}{c|ccc|} & ab & ac & bc \\ \hline a & 1 & 1 & 0 \\ b & 1 & 0 & 1 \\ c & 0 & 1 & 1 \\ \hline \end{array}$$

The incidence matrix in this case has determinant -2, hence it is not totally unimodular.

3.3 Further results on totally unimodular matrices

Next we report further results relating to totally unimodularity that are often useful in proving that certain matrices are t.u., or that certain formulations are perfect.

Theorem 3.7. *Let A be a matrix with all entries in $\{0, 1, -1\}$, such that in every column of A there is exactly one entry of value 1, one entry of value -1, and all other entries with value 0. Then A is totally unimodular.*

The proof is almost identical to the proof of Theorem 3.6, requiring only a minor modification at the end, so we only sketch it here.

Sketch of proof. Suppose by contradiction that there is a submatrix with determinant different from $0, 1, -1$. Among all such matrices, let B be the one with the smallest possible number of rows. Each column of the matrix B has at most two nonzero entries. We can rule out the case that B contains a column with less than two nonzero entries by the same arguments we used in cases a) and b) in the proof of Theorem 2.4. Thus every column of B has exactly a $+1$ entry and a -1 entry. It follows that the sum of all rows of B is the 0 vector. Hence $\det(B) = 0$, a contradiction. $\square$

Theorem 3.8. *Let $A \in \mathbb{R}^{m\times n}$ be a totally unimodular matrix. Given a vector $b \in \mathbb{Z}^m$, all vertices of the polyhedron $\{x \in \mathbb{R}^n \mid Ax \le b\}$ are integer.*

Proof. Let $\bar{x}$ be a vertex of the polyhedron $P = \{x \in \mathbb{R}^n \mid Ax \le b\}$. Since $\bar{x}$ is a vertex, there exist n linearly independent constraints of the system $Ax \le b$ that are binding at $\bar{x}$. Let $\bar{A}x \le \bar{b}$ be these n linearly independent constraints. Thus $\bar{A}$ is an $n \times n$ nonsingular matrix and $\bar{x} = \bar{A}^{-1}\bar{b}$. Since A is totally unimodular, it follows from Observation 3.4 that $\bar{A}^{-1}$ is an integer matrix. Since b is an integer vector, it then follows that $\bar{x}$ is integer. $\square$

Theorem 3.9. *Let $A \in \mathbb{R}^{m\times n}$ be a totally unimodular matrix. The following hold*

i) $A^\top$ *is T.U.*

ii) If matrix A' is obtained from A by multiplying one row or column by -1, then A' is T.U.

iii) The matrix $(A|-A)$, obtained by juxtaposing the matrices A and $-A$, is T.U.

iv) Let $u \in \mathbb{Z}^m$ be a unit vector (i.e. all entries are 0 except for one, with value 1). Then the $m \times (n+1)$-matrix $(A|u)$ obtained by juxtaposing the matrix A and the column vector u is T.U.

v) The matrix $(A|I)$, obtained by juxtaposing the matrix A and the identity matrix I, is T.U.

Proof. i) This is obvious, since transposing a matrix does not change the determinant.
ii) If B is a square submatrix of A', then either it is a submatrix of A or it is obtained by a submatrix B' of A by multiplying one row or column by -1. Multiplying a row or column of a square matrix only changes the sign of the determinant. Thus $\det(B) = \pm\det(B)$.
iii) Let $A_1, \ldots, A_n$ be the columns of A. If a square submatrix B of $(A| - A)$ contains both columns A_j and $-A_j$ for some index j, then B is singular. If the matrix B contains at most one of A_j or $-A_j$ for every $j = 1, \ldots, n$, then B is obtained from a submatrix of A by multiplying by -1 some of the columns. By point 3., it follows that $\det(B) \in \{0, 1, -1\}$.
iv) Given a square submatrix B of $(A|u)$, either B is a submatrix of A, in which case $\det(B) \in \{0, 1, -1\}$, or B contains part of the new column u. If this part of the new column does not contain the 1, then B contains a column with all zeros, so $\det(B) = 0$. Otherwise B contains a unit vector, and the matrix B' obtained from B by deleting this unit column and the row where there is a 1 in the unit column is a submatrix of A. Thus $\det(B') \in \{0, 1, -1\}$ and $\det(B) = \pm\det(B')$. (See Figure 3.3 for a depiction of the three cases)

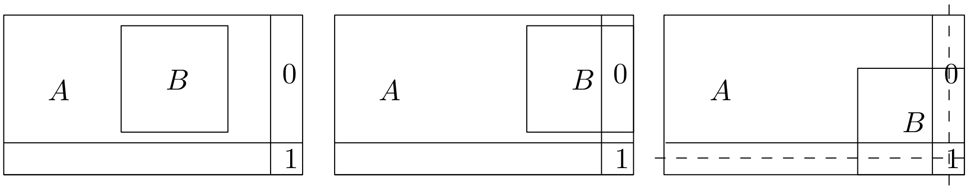


Figure 3.3: Three cases in the proof of iv).

v) This follows by repeatedly applying iv), starting from A and juxtaposing a column of I at the time. □

Theorems 3.8 and 3.9 imply the following.

Corollary 3.10. *Let $A \in \mathbb{R}^{m\times n}$ be a totally unimodular matrix. Given vectors $b, d \in \mathbb{Z}^m$ and $\ell, u \in \mathbb{Z}^n$ all vertices of the polyhedron*

$$\{x \in \mathbb{R}^n \mid b \le Ax \le d,\ \ell \le x \le u\}$$

are integer.

Proof. We can write the constraints of the above problem in the following form

$$\begin{pmatrix} A \\ -A \\ I \\ -I \end{pmatrix} x \le \begin{pmatrix} d \\ -b \\ u \\ -\ell \end{pmatrix}$$

By Theorem 3.9, the constraint matrix of the above system is totally unimodular. Since the right-hand-side vector is integer, it follows from Theorem 3.8 that all the vertices of the polyhedron defined by the above constraints are integer. □

Ghouila-Houri's theorem Next we explain a theorem that provides necessary and sufficient conditions for a matrix to be totally unimodular. Given an $m \times n$ matrix A with entries in $\{0, 1, -1\}$, a *column bicoloring* of A is a partition of its column set $\{1, \ldots, n\}$ into two classes (R, B), say red and blue, $R \cup B = \{1, \ldots, n\}$, $R \cap B = \emptyset$. We say that a column bicoloring is an *equitable bicoloring* if, for every row, the sum of the red entries in the row differs by at most one from the sum of the blue entries. Formally, the bicoloring (R, B) is equitable if, for all $i \in \{1, \ldots, m\}$,

$$\left| \sum_{j\in R} a_{ij} - \sum_{j\in B} a_{ij} \right| \le 1.$$

Similarly, a *row bicoloring* of A is a column bicoloring of $A^\top$, and a row bicoloring of A is equitable if it is an equitable column bicoloring of $A^\top$.

For example, below we show a matrix with an equitable bicoloring

R	B	R	B	R
1	1	1	0	0
0	0	1	1	1
0	1	1	0	0
−1	0	1	0	−1

Theorem 3.11 (Ghouila-Houri, 1962). *Let A be a matrix with entries in $\{0, 1, -1\}$. The following are equivalent*

i) A is totally unimodular.

ii) Every submatrix of A has an equitable column bicoloring.

iii) Every submatrix of A has an equitable row bicoloring.

Proof of necessity. We do not provide a complete proof here, but only prove the "easy direction", namely, that *i*) implies *ii*). Note that this also provides a proof that *i*) implies *iii*), since we know that A is totally unimodular if and only if $A^\top$ is totally unimodular. To show that *i*) implies *ii*), assume A is totally unimodular. It suffices to shows that A has an equitable bicoloring, because all submatrices of A are also totally unimodular. Consider now the polyhedron P defined by

$$P = \{x \in \mathbb{R}^n \mid \ell \leq Ax \leq u,\, 0 \leq x \leq \mathbf{1}\},$$

where $\ell, u \in \mathbb{R}^m$ are defined by

$$\ell_i = \left\lfloor \sum_{j=1}^{n} \frac{a_{ij}}{2} \right\rfloor, \quad u_i = \left\lceil \sum_{j=1}^{n} \frac{a_{ij}}{2} \right\rceil \qquad \text{for } i = 1, \ldots, m.$$

(That is, ℓ is the vector obtained by taking the sum of all columns of A, dividing all entries by 2, and rounding down all entries, whereas u is the vector obtained by taking the sum of all columns of A, dividing all entries by 2, and rounding up all entries.) Observe that P is nonempty, because the vector $\bar{x} \in \mathbb{R}^n$ with entries all equal to $1/2$ is in P. Since A is a totally unimodular matrix and ℓ, u have integer components, it follows from Corollary 3.10 that P has only integer vertices. In particular, all vertices must have $0, 1$ components, and there must be one such vertex v because P is nonempty. Let us now color red the columns of A corresponding to entries of v with value 1, and blue the columns of A corresponding to entries with value 0. It is immediate to check that such bicoloring is equitable. □

For example, the matrix

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

is not totally unimodular, since it does not have an equitable column bicoloring. Indeed, if we assign colour red to the first column, then the first rows forces the colour of the second column to be blue, and the second row forces the colour of the third column to be blue. However, now the sum of the red entries in the third row is zero, and the sum of the blue entries in the third row is 2.

3.4 An ideal formulation for the matching problem in general graphs

Recall that the formulation (3.1) is not sufficient to describe the convex hull of the integer feasible solution. What are the conditions not captured by the degree constraints in these example?

Consider a general graph $G = (V, E)$ and a set of nodes $U \subseteq V$ of odd cardinality. Let $E[U] := \{uv \in E \,|\, u, v \in U\}$; that is, $E[U]$ is the set of edges of G with both endnodes in U. Given a matching M of G, every edge of M that has both endnodes in U covers exactly two elements of U. Since the edges in M are pairwise disjoint, it follows that $2|M \cap E[U]| \leq |U|$. Thus $|M \cap E[U]| \leq |U|/2$. Since $|M|$ is integer, it follows that $|M \cap E[U]| \leq \lfloor |U|/2 \rfloor$. Since $|U|$ is odd, $\lfloor |U|/2 \rfloor = (|U| - 1)/2$. Therefore, for every matching M, the number of edges in M with both endnodes in U is at most $(|U| - 1)/2$.

This means that, if x is the characteristic vector of some matching, x must satisfy the inequality

$$\sum_{e \in E[U]} x_e \leq \frac{|U| - 1}{2}.$$

In the above example, the inequality relative to the odd set $\{1, 2, 3\}$ would be $x_{12} + x_{13} + x_{23} \leq 1$. Adding this inequality cuts off the point $(1/2, 1/2, 1/2)$.

Adding the above family of inequalities to the initial formulation, we get the following new, tighter formulation

$$\begin{array}{rcll} \max \sum_{e \in E} x_e & & & \\ \sum_{uv \in E} x_{uv} & \leq & 1, & v \in V \\ \sum_{e \in E[U]} x_e & \leq & \frac{|U|-1}{2} & U \subseteq V,\ |U| \text{ odd}, \\ x_e & \geq & 0 & e \in E. \end{array} \tag{3.3}$$

A classic theorem of Edmonds (1965) states that the above is indeed the ideal formulation for the matching problem. That is, all the vertices of the feasible region have components in $\{0, 1\}$.

Chapter 4

The cutting planes method

In general, there is not hope of knowing the ideal formulation for the MILP we would like to solve. One might, however, try to start with some initial formulation, solve it, and then try to tighten it by adding a one of more inequalities and repeat the process, thus getting linear relaxation that approximate better and better the convex hull of the feasible region around the optimal solution. This is the basic idea behind cutting plane algorithms.

More formally, suppose we want to solve a given MILP problem

$$\begin{array}{c} \max c^\top x \\ x \in X \end{array} \qquad (P_I) \tag{4.1}$$

where the feasible region $X = \{x \in \mathbb{R}^n \mid Ax \leq b,\, x \geq 0,\, x_i \in \mathbb{Z} \text{ for all } i \in I\}$ is defined by a given matrix $A \in \mathbb{Q}^{m\times n}$ and vector $b \in \mathbb{Q}^m$.

We say that a linear inequality $\alpha^\top x \leq \beta$, (where $\alpha \in \mathbb{R}^n$, $\beta \in \mathbb{R}$) is *valid* for X if, for all $x \in X$, $\alpha^\top x \leq \beta$ holds.

In particular, if we are given any formulation $A'x \leq b'$, $x \geq 0$, for X, and if $\alpha^\top x \leq \beta$ is a valid inequality for X, then the system

$$\begin{array}{c} Ax \leq b \\ \alpha^\top x \leq \beta \\ x \geq 0 \end{array} \tag{4.2}$$

is also a formulation for X.

Given any point for $x^* \notin \text{conv}(X)$, we say that $\alpha^\top x \leq \beta$ *cuts-off* x^* if $\alpha^\top x^* > \beta$. If x^* is a point satisfying $A'x^* \leq b'$, $x^* \geq 0$ that is cut-off by x^*, then the formulation (4.2) is tighter than the original formulation (4.1).

Cutting planes method

Define as initial LP relaxation the problem $\max\{c^\top x \mid Ax \leq b,\, x \geq 0\}$.

1. Solve the current relaxation, and let x^* be the optimal solution found;
2. If $x^* \in X$, then x^* is an optimal solution to the MILP (4.1), STOP.
3. Otherwise, find a valid inequality $\alpha^\top x \leq \beta$ for X cutting-off x^*;
4. Add the constraint $\alpha^\top x \leq \beta$ to the current linear relaxation and return to 1.

Obviously, the cutting plane method described here is a general technique to tackle MILP problems, but to implement it is necessary to device an automatic way to generate valid inequalities cutting-off the current fractional solution. Such a valid inequality is usually referred to as a *cutting plane*.

4.1 Gomory cuts

We illustrate here a method to generate cutting planes. Note that such method, due to Ralph Gomory, is applicable only to pure integer programs. There are similar methods that can be adopted to general mixed-integer linear programs, but we do not cover them in these notes.

For a real number $x \in \mathbb{R}$, we use $\lfloor x \rfloor$ to denote the integer part, i.e. the largest integer smaller or equal to x. Note that $\lfloor 1.5 \rfloor = 1$, whereas $\lfloor -1.5 \rfloor = -2$.

Consider a pure integer programming problem in the following form:

$$\begin{aligned} z_I = \max c^\top x \\ Ax = b \\ x \geq 0 \\ x \in \mathbb{Z}^n \end{aligned} \tag{4.3}$$

where $A \in \mathbb{Q}^{m\times n}$ and $b \in \mathbb{Q}^m$. Let $X = \{x \,|\, Ax = b,\, x \geq 0,\, x \in \mathbb{Z}^n\}$ be the set of feasible points.

Solving the linear relaxation of the above problem, we end up with an optimal basic feasible solution x^* relative to some optimal basis B (as usual we denote by N the set of indices of non-basic variables). The problem in tableau form with respect to the basis B is

$$\begin{array}{llrcll} \max & z & & & & \\ & z & -\sum_{j\in N} \bar{c}_j x_j & = & z_B & \\ & & x_{\beta[i]} \quad +\sum_{j\in N} \bar{a}_{ij} x_j & = & \bar{b}_i, & i = 1,\dots,m \\ & & x & \geq & 0. & \end{array}$$

Since B is an optimal basis, all reduced costs are nonpositive, i.e. $\bar{c}_j \leq 0$, $j \in N$. The optimal solution x^* is defined by

$$\begin{array}{rcll} x^*_{\beta[i]} & = & \bar{b}_i, & i = 1,\dots,m; \\ x^*_j & = & 0, & j \in N; \end{array}$$

hence x^* is feasible for the original integer programming problem (4.3) if and only if $\bar{b}_i \in \mathbb{Z}$ for $i = 1,\dots,m$. If this happens, then x^* is an optimal solution to (4.3)

If not, then there is some index $h \in \{1,\dots,m\}$ such that $\bar{b}_h \notin \mathbb{Z}$.

Now, any point x satisfying the constraints $Ax = b$, $x \geq 0$, should also satisfy

$$x_{\beta[h]} + \sum_{j\in N} \lfloor \bar{a}_{hj} \rfloor x_j \leq \bar{b}_h$$

because

$$\bar{b}_h = x_{\beta[h]} + \sum_{j\in N} \bar{a}_{hj} x_j \geq x_{\beta[h]} + \sum_{j\in N} \lfloor \bar{a}_{hj} \rfloor x_j$$

where the inequality holds because $x_j \geq 0$ and $\bar{a}_{hj} \geq \lfloor \bar{a}_{hj} \rfloor$ for all j.

Now, since (4.3) is a pure integer program, any integer point x satisfying the constraints $Ax = b$, $x \geq 0$ must satisfy

$$x_{\beta[h]} + \sum_{j\in N} \lfloor \bar{a}_{hj} \rfloor x_j \leq \lfloor \bar{b}_h \rfloor \tag{4.4}$$

since $x_{\beta[h]} + \sum_{j\in N} \lfloor \bar{a}_{hj} \rfloor x_j$ is an integer number, and any integer number less than or equal to $\bar{b}_h$ cannot exceed $\lfloor \bar{b}_h \rfloor$.

The inequality (4.4) is said *Gomory cut*. From the above discussion, the Gomory cut (4.4) is valid for X, and it cuts off the current (fractional) solution x^* because

$$x^*_{\beta[h]} + \sum_{j\in N} \lfloor \bar{a}_{hj} \rfloor x^*_j = x^*_{\beta[h]} = \bar{b}_h > \lfloor \bar{b}_h \rfloor$$

(note that $\bar{b}_h > \lfloor \bar{b}_h \rfloor$ because $\bar{b}_h$ is not an integer number).

It is convenient to express the Gomory cut (4.4) in equivalent form. Adding a slack variable s, the Gomory cut (4.4) becomes

$$x_{\beta[h]} + \sum_{j \in N} \lfloor \bar{a}_{hj} \rfloor x_j + s = \lfloor \bar{b}_h \rfloor, \quad s \geq 0.$$

Note that, since all coefficients in the above equation are integer, then if x is an integer vector also $s = \lfloor \bar{b}_h \rfloor - x_{\beta[h]} - \sum_{j \in N} \lfloor \bar{a}_{hj} \rfloor x_j$ must be integer, and we may therefore impose that s is also an integer variable (in other words, even after we add the new variable s we still keep the problem a pure integer one).

Subtracting the previous equation from the row of the tableau

$$x_{\beta[h]} + \sum_{j \in N} \bar{a}_{hj} x_j = \bar{b}_h,$$

we obtain the equation

$$\sum_{j \in N} (\lfloor \bar{a}_{hj} \rfloor - a_{hj}) x_j + s = \lfloor \bar{b}_h \rfloor - \bar{b}_h.$$

This is the so called *Gomory cut in fractional form.*

Adding the above equation to the previous tableau, we obtain the problem

$$\begin{array}{lllllll}
\max & z & & & & & \\
 & -z & & +\sum_{j \in N} \bar{c}_j x_j & & = & -z_B \\
 & & x_{\beta[i]} & +\sum_{j \in N} \bar{a}_{ij} x_j & & = & \bar{b}_i, \qquad i = 1, \ldots, m \\
 & & & \sum_{j \in N} (\lfloor \bar{a}_{hj} \rfloor - a_{hj}) x_j & +s & = & \lfloor \bar{b}_h \rfloor - \bar{b}_h \\
 & & & x, & s & \geq & 0.
\end{array}$$

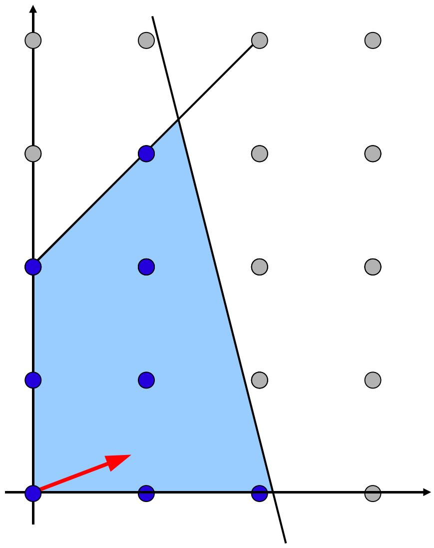
Note that the above problem is already in tableau form, where the basic variables are $x_{\beta[1]}, \ldots, x_{\beta[m]}, s$. Furthermore, the basis is dual feasible because the reduced costs are all nonpositive (since $c_j \leq 0$ for all $j \in N$). Note that the right-hand-side of the new constraint (relative to the basic variable s) is $\lfloor \bar{b}_h \rfloor - \bar{b}_h < 0$, hence the basis is not primal feasible.

We can therefore apply the dual simplex method to solve the new problem to optimality starting from the current dual feasible basis rather than restarting from scratch. In implementations of the cutting planes method this fact is of fundamental importance, since empirical studies find that typically much fewer iterations of the dual simplex are necessary to re-establish optimality if we start from the current basis.

Example.

Consider the pure integer programming problem

$$\begin{aligned} \max\ z &= 11x_1 + 4.2x_2 \\ -x_1 + x_2 &\le 2 \\ 8x_1 + 2x_2 &\le 17 \\ x_1, x_2 &\ge 0 \text{ integer.} \end{aligned} \tag{4.5}$$



Adding slack variables x_3 and x_4 we transform the problem in standard form:

$$\begin{aligned} z - 11x_1 - 4.2x_2 &= 0 \\ -x_1 + x_2 + x_3 &= 2 \\ 8x_1 + 2x_2 + x_4 &= 17 \\ x_1, x_2, x_3, x_4 &\ge 0 \text{ integer.} \end{aligned}$$

(Note that, since the constraint coefficients of x_1 and x_2 are all integer, then $x_3 = 2 + x_1 - x_2$ and $x_4 = 17 - 8x_1 - 2x_2$ are integer whenever x_1 and x_2 are integer, thus we could impose the condition that x_3 and x_4 are integer variables).

After solving the linear relaxation, we obtain the tableau

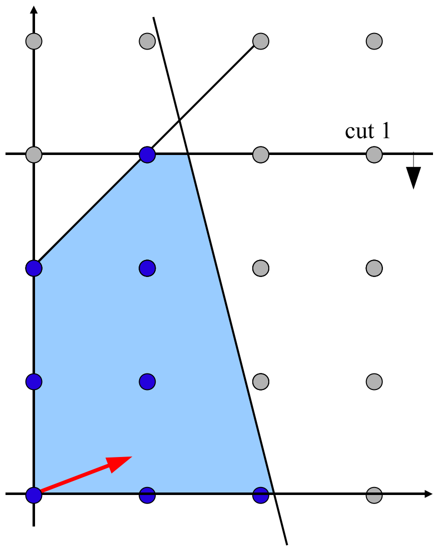
$$\begin{array}{ccccccc} z & & & +1.16x_3 & +1.52x_4 & = & 28.16 \\ & & x_2 & +0.8x_3 & +0.1x_4 & = & 3.3 \\ & x_1 & & -0.2x_3 & +0.1x_4 & = & 1.3 \end{array}$$

The corresponding optimal solution is $x_3 = x_4 = 0$, $x_1 = 1.3$, $x_2 = 3.3$ with objective value $z = 28.16$. Since x_1 and x_2 take fractional values, the current solution is not feasible for the integer programming problem (4.5). We can generate the Gomory cut from the equation $x_2 + 0.8x_3 + 0.1x_4 = 3.3$, namely

$$x_2 \le 3$$

Adding the above cut to the previous formulation, we obtain the new relaxation

$$\begin{aligned} &\max\ 11x_1 + 4.2x_2 \\ &\quad -x_1 + x_2 \le 2 \\ &\quad 8x_1 + 2x_2 \le 17 \\ &\quad x_2 \le 3 \\ &\quad x_1, x_2 \ge 0. \end{aligned}$$



To re-optimize, we write the Gomory cut in fractional form, adding the slack variable x_5:

$$-0.8x_3 - 0.1x_4 + x_5 = -0.3.$$

Adding to the previous tableau, we obtain the new tableau

$$\begin{array}{cccccccc}
z & & & +1.16x_3 & +1.52x_4 & & = & 28.16 \\
& & x_2 & +0.8x_3 & +0.1x_4 & & = & 3.3 \\
& x_1 & & -0.2x_3 & +0.1x_4 & & = & 1.3 \\
& & & -0.8x_3 & -0.1x_4 & +x_5 & = & -0.3
\end{array}$$

relative to the basis $\{1, 2, 5\}$. The basis is dual feasible but not primal feasible. Applying the dual simplex method, we select x_5 as exiting variable, while the entering variable is x_3 since $\min\{1.16/0.8, 1.52/0.1\} = 1.16/0.8$.

The new tableau is

$$\begin{array}{cccccccc}
z & & & & +1.375x_4 & +1.45x_5 & = & 27.725 \\
& & x_2 & & & +x_5 & = & 3 \\
& x_1 & & & +0.125x_4 & -0.25x_5 & = & 1.375 \\
& & & x_3 & +0.125x_4 & -1.25x_5 & = & 0.375
\end{array}$$

which is optimal. The new basic optimal solution is $x_1 = 1.375$, $x_2 = 3$ $x_3 = 0.375$, with value $z = 27.725$. The solution is not integer.

From the tableau equation $x_3 + 0.125x_4 - 1.25x_5 = 0.375$ we generate the Gomory cut

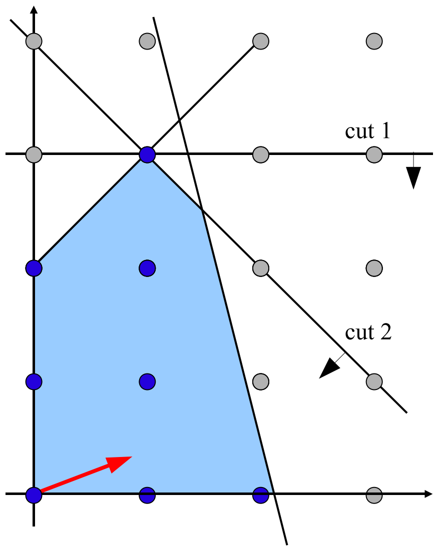
$$x_3 - 2x_5 \leq 0.$$

To represent the cut in the original space of variables, we can write the cut in terms of x_1 and x_2 (since $x_3 = 2 + x_1 - x_2$ and $x_5 = 3 - x_2$), obtaining

$$x_1 + x_2 \leq 4.$$

Hence the current linear relaxation is now

$$\begin{aligned}
\max\ & 11x_1 + 4.2x_2 \\
& -x_1 + x_2 \leq 2 \\
& 8x_1 + 2x_2 \leq 17 \\
& x_2 \leq 3 \\
& x_1 + x_2 \leq 4 \\
& x_1, x_2 \geq 0.
\end{aligned}$$



The Gomory cut in fractional form is

$$-0.125x_4 - 0.75x_5 + x_6 = -0.375$$

and the new tableau is

$$\begin{array}{rrrrrrcl}
-z & & & +1.375x_4 & +1.45x_5 & & = & 27.725 \\
& x_2 & & & +x_5 & & = & 3 \\
x_1 & & & +0.125x_4 & -0.25x_5 & & = & 1.375 \\
& & x_3 & +0.125x_4 & -1.25x_5 & & = & 0.375 \\
& & & -0.125x_4 & -0.75x_5 & +x_6 & = & -0.375
\end{array}$$

Applying the dual simplex, the variable x_6 leaves the basis while the variable x_5 enters. The new tableau is

$$\begin{array}{rrrrrrcl}
z & & & +17/15x_4 & & +29/15x_6 & = & 27 \\
& x_2 & & -1/6x_4 & & +4/3x_6 & = & 2.5 \\
x_1 & & & +1/6x_4 & & -1/3x_6 & = & 1.5 \\
& & x_3 & +1/3x_4 & & -5/3x_6 & = & 1 \\
& & & 1/6x_4 & +x_5 & -4/3x_6 & = & 0.5
\end{array}$$

which is optimal.

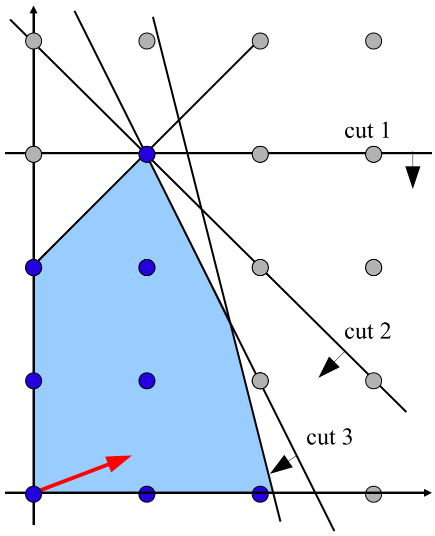
The new optimal solution is $x_1 = 1.5$, $x_2 = 2.5$ with value $z = 27$. Again, the solution is not integer.

From the tableau equation $1/6x_4 + x_5 - 4/3x_6 = 0.5$ we generate the Gomory cut $x_5 - 2x_6 \leq 0$ which in the space of the variables x_1, x_2 can be written as $2x_1 + x_2 \leq 5$ (since $x_5 = 3 - x_2$ and $x_6 = 4 - x_1 - x_2$). The new linear relaxation is

max $11x_1 + 4.2x_2$
$-x_1 + x_2 \leq 2$
$8x_1 + 2x_2 \leq 17$
$x_2 \leq 3$
$x_1 + x_2 \leq 4$
$2x_1 + x_2 \leq 5$
$x_1, x_2 \geq 0.$

The Gomory cut in fractional form is

$$-1/6x_4 - 2/3x_6 + x_7 = -0.5$$



Adding it to the previous tableau, we get the new one

$$\begin{array}{cccccccccc}
z & & & & +17/15x_4 & & +29/15x_6 & & = & 27 \\
& & x_2 & & -1/6x_4 & & +4/3x_6 & & = & 2.5 \\
& x_1 & & & +1/6x_4 & & -1/3x_6 & & = & 1.5 \\
& & & x_3 & +1/3x_4 & & -5/3x_6 & & = & 1 \\
& & & & 1/6x_4 & +x_5 & -4/3x_6 & & = & 0.5 \\
& & & & -1/6x_4 & & -2/3x_6 & +x_7 & = & -0.5
\end{array}$$

If we apply the dual simplex method x_7 leaves the basis and x_6 enters. The new tableau is

$$\begin{array}{cccccccccc}
z & & & & +13/20x_4 & & & 29/10x_7 & = & 25.55 \\
& & x_2 & & -1/2x_4 & & & +2x_7 & = & 3/2 \\
& x_1 & & & +1/4x_4 & & & -1/2x_7 & = & 7/4 \\
& & & x_3 & +3/4x_4 & & & -5/2x_7 & = & 9/4 \\
& & & & 1/2x_4 & +x_5 & & -2x_7 & = & 3/2 \\
& & & & 1/4x_4 & & x_6 & -3/2x_7 & = & 3/4
\end{array}$$

which is optimal

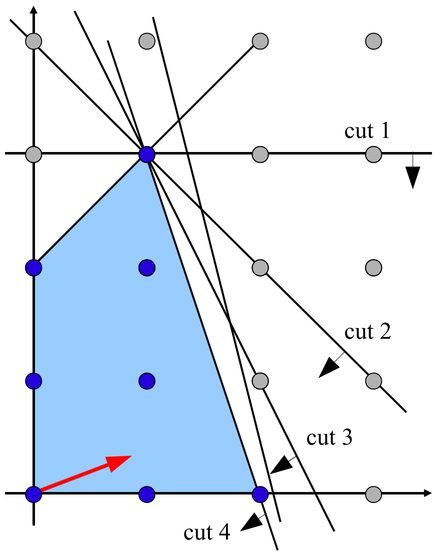
The new optimal solution is $x_1 = 1.75$, $x_2 = 1.5$ with value $z = 25.5$. Again, the solution is not integer.

From the tableau equation $x_1 + 1/4x_4 - 1/2x_7 = 7/4$ we generate the Gomory cut $x_1 - x_7 \leq 1$ which in the space of the variables x_1, x_2 can be written as $3x_1 + x_2 \leq 6$ (since $x_6 = 5 - 2_1 - x_2$). The new linear relaxation is

max $11x_1 + 4.2x_2$
$-x_1 + x_2 \leq 2$
$8x_1 + 2x_2 \leq 17$
$x_2 \leq 3$
$x_1 + x_2 \leq 4$
$2x_1 + x_2 \leq 5$
$3x_1 + x_2 \leq 6$
$x_1, x_2 \geq 0.$

The Gomory cut in fractional form is

$$-1/4x_4 - 1/2x_7 + x_8 = -3/4$$



Adding it to the previous tableau, we get the new one

$$\begin{array}{ccccccccccc}
z & & & & +13/20x_4 & & & 29/10x_7 & & = & 25.55 \\
 & & x_2 & & -1/2x_4 & & & +2x_7 & & = & 3/2 \\
 & x_1 & & & +1/4x_4 & & & -1/2x_7 & & = & 7/4 \\
 & & & x_3 & +3/4x_4 & & & -5/2x_7 & & = & 9/4 \\
 & & & & 1/2x_4 & +x_5 & & -2x_7 & & = & 3/2 \\
 & & & & 1/4x_4 & & x_6 & -3/2x_7 & & = & 3/4 \\
 & & & & -1/4x_4 & & & -1/2x_7 & +x_8 & = & -3/4
\end{array}$$

If we apply the dual simplex method x_8 leaves the basis and x_4 enters. The new tableau is

$$\begin{array}{ccccccccccc}
z & & & & + & & & 8/5x_7 & +13/5x_8 & = & 23.6 \\
 & & x_2 & & & & & +3x_7 & -2x_8 & = & 3 \\
 & x_1 & & & & & & -x_7 & +x_8 & = & 1 \\
 & & & x_3 & & & & -4x_7 & +3x_8 & = & 0 \\
 & & & & & +x_5 & & -3x_7 & +2x_8 & = & 0 \\
 & & & & & & x_6 & -2x_7 & +x_8 & = & 0 \\
 & & & & x_4 & & & +2x_7 & -4x_8 & = & 3
\end{array}$$

which is optimal.

The optimal solution is $x_1 = 1$, $x_2 = 3$, with value $z = 23.6$. Since the solution is integer, we have found the solution of original integer program, namely the point $(1, 3)$.

4.2 Branch-and-cut

State-of-the-art solvers employ the so called *branch-and-cut* method, which is a combination of branch-and-bound and cutting planes. Branch-and-cut differs from branch-and-bound as described in Table 1.1 in that, at point (d), rather than immediately branching on some fractional variable, one might decide to add one or more cutting planes to tighten the formulation of the subproblem at the current node.

The decision of whether or not to add cuts is made empirically, typically on the success of previously added cuts. Most commonly, several rounds of cuts are added at the root node, while fewer or no cuts are generated deeper in the tree.

We remark that over the last decade there has been a dramatic improvement of MILP solvers: not only have solvers become several order of magnitude faster, but many problems that up to the late '90s were computationally intractable are now well within reach of state-of-the art solvers. This improvement is due to several factors – not least better hardware and faster LP solvers – but a central role has been the introduction in commercial solvers of cutting planes. Even though the idea of cutting planes dates back to the '60s and it had been successfully adopted in specific applications such as the Traveling Salesman Problem, it was not until the mid '90s that the practical importance of cutting planes in general purpose solvers was recognized. State-of-the-art solvers implement an arsenal of cuts, including Gomory Mixed Integer Cuts (a generalization of the Gomory fractional cuts we discussed in these notes) and knapsack cover inequalities.

The effectiveness and reliability of current MILP solvers has made Integer Programming – once regarded as a desperately difficult problem, useful only in few lucky instances – a practical decision tool in the most diverse areas.