6. INTEGER PROGRAMMING

1. Some applications of integer programming

2. Solving integer programming problems

3. The graphical solution of integer programming problems

4. The branch and bound method

   4.1 The branch and bound algorithm

5. 0-1 integer programming

   5.1 A 0-1 branch and bound algorithm

Integer Programming

Integer programming (IP) deals with solving linear models in which some or all the variables are restricted to be integer.

IP problems are usually much more difficult to solve than LP problems.

Three types of IP models:

- In mixed integer programming, only some of the variables are restricted to integer values.

- In pure integer programming, all the variables are integers.

- In binary integer programming or 0-1 integer programming, all the variables are binary (restricted to the values 0 or 1).

1. Some applications of integer programming

Example 1. The number of employees needed:

Day	Employees
1. Monday	15
2. Tuesday	13
3. Wednesday	15
4. Thursday	18
5. Friday	14
6. Saturday	16
7. Sunday	10

Employees work five consecutive days and have the next two days off.

The objective is to employ the minimum number of workers. Decision variables:

x_j : number of employees whose working shift starts on day j, $\quad j = 1, \ldots, 7$.

$$\min \ z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$$

subject to

$$x_1 + x_4 + x_5 + x_6 + x_7 \geq 15$$
$$x_1 + x_2 + x_5 + x_6 + x_7 \geq 13$$
$$x_1 + x_2 + x_3 + x_6 + x_7 \geq 15$$
$$x_1 + x_2 + x_3 + x_4 + x_7 \geq 18$$
$$x_1 + x_2 + x_3 + x_4 + x_5 \geq 14$$
$$x_2 + x_3 + x_4 + x_5 + x_6 \geq 16$$
$$x_3 + x_4 + x_5 + x_6 + x_7 \geq 10$$
$$x_1, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0 \text{ and integer}$$

Example 2. A knapsack problem

We want to put four items in a knapsack that can hold up to 12 kg. The weight and the value associated to each of the items:

	1	2	3	4
Weight (kg)	3	6	5	5
Value (euro)	15	25	12	10

We need to decide which items to put in so as to maximize the total value of the knapsack.

We define four binary variables, one for each item j, $j = 1, 2, 3, 4$:

$$x_j = \begin{cases} 1 & \text{if item } j \text{ is introduced in the knapsack} \\ 0 & \text{otherwise} \end{cases}$$

The 0-1 IP model that represents the problem is:

$$\begin{aligned} \max \ z = {} & 15x_1 + 25x_2 + 12x_3 + 10x_4 \\ & \text{subject to} \\ & 3x_1 + 6x_2 + 5x_3 + 5x_4 \leq 12 \\ & x_1, x_2, x_3, x_4 = 0 \text{ or } 1 \end{aligned}$$

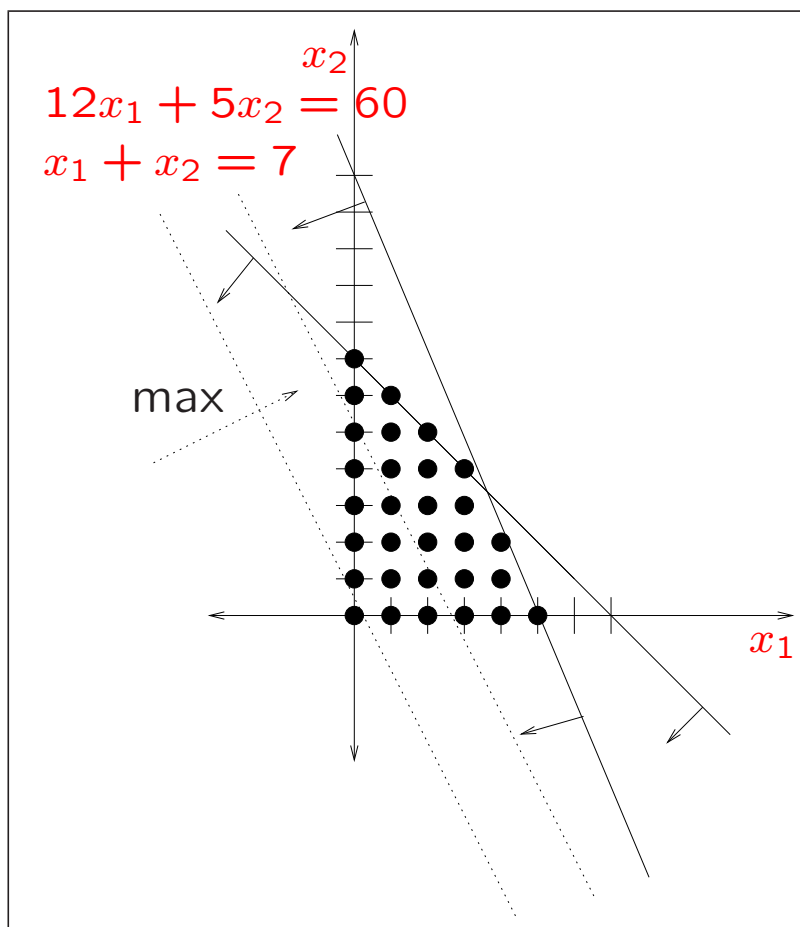# 2. Solving integer programming problems

We illustrate by means of an example the difficulties found while solving an IP problem.
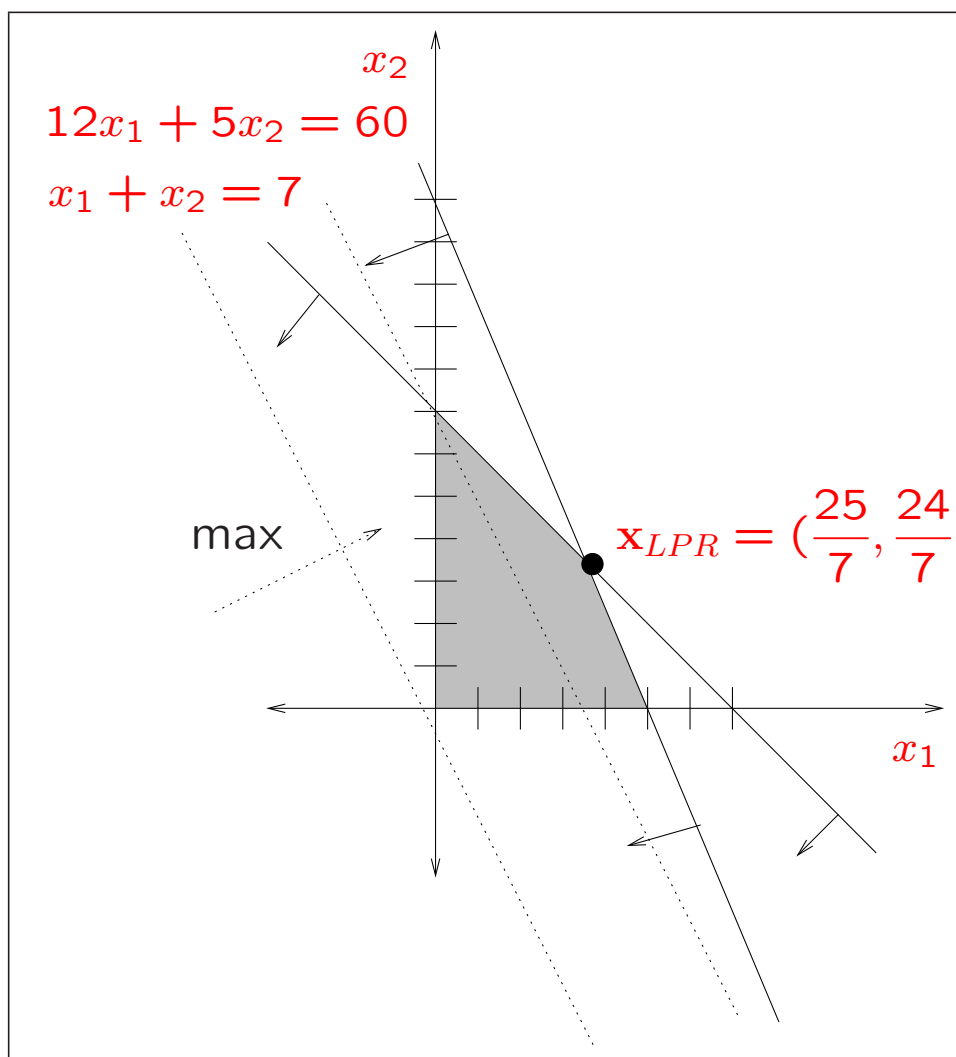
Example:

$$\max \ z = 80x_1 + 45x_2$$
$$\text{subject to}$$
$$x_1 + x_2 \leq 7$$
$$12x_1 + 5x_2 \leq 60$$
$$x_1, x_2 \geq 0 \text{ and integer}$$

The graphical representation of the problem shows the set of solutions:

The LP problem obtained by ignoring the integer constraints is called its LP relaxation.

The optimal solution to the IP problem may be found throught the solution of a sequence of LP relaxation problems.

3. The graphical solution of IP problems

The optimal solution to the LP relaxation (LPR) is:

$$\mathbf{x}_{LPR} = (3.571, 3.428)$$

Problem LP2

max $z = 80x_1 + 45x_2$
subject to

$$x_1 + x_2 \le 7$$
$$12x_1 + 5x_2 \le 60$$
$$x_1 \le 3$$
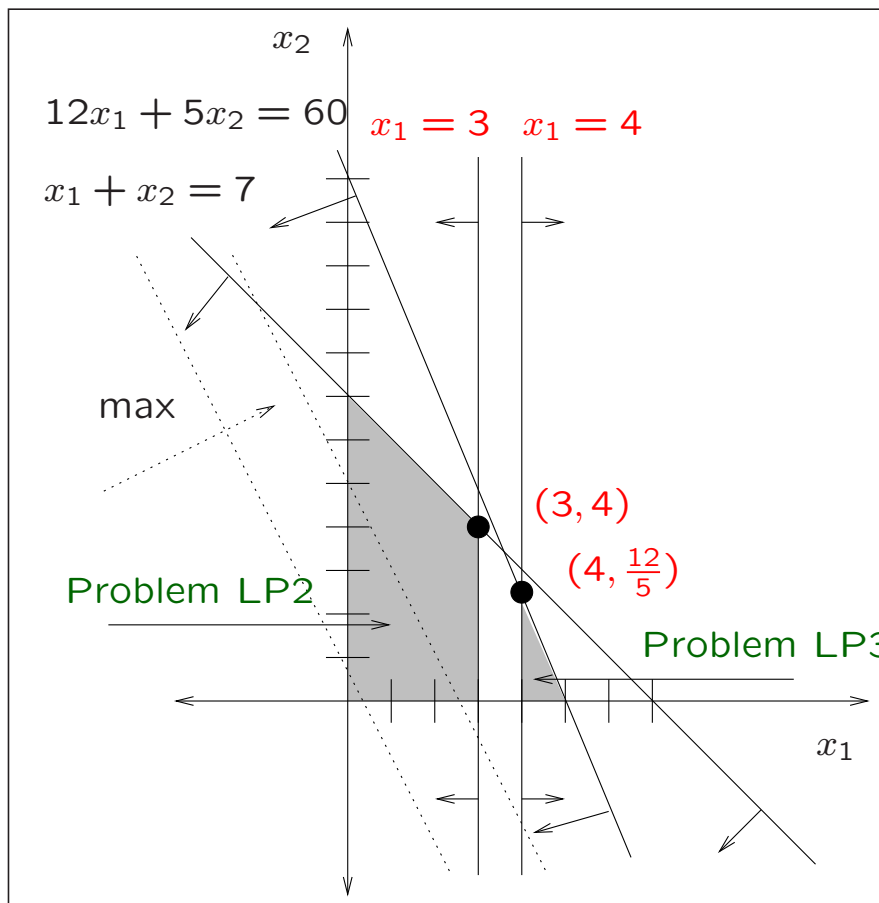$$x_1, x_2 \ge 0$$

Problem LP3

max $z = 80x_1 + 45x_2$
subject to

$$x_1 + x_2 \le 7$$
$$12x_1 + 5x_2 \le 60$$
$$x_1 \ge 4$$
$$x_1, x_2 \ge 0$$

LP2 is pruned. $\mathbf{x}_{LP2}$ candidate solution $\to z_{LB} = 420$.

LP3 is not pruned: $z_{LP3} = 428$.

Problem LP4

max $z = 80x_1 + 45x_2$

subject to

$$x_1 + x_2 \leq 7$$
$$12x_1 + 5x_2 \leq 60$$
$$x_1 \geq 4, \ x_2 \leq 2$$
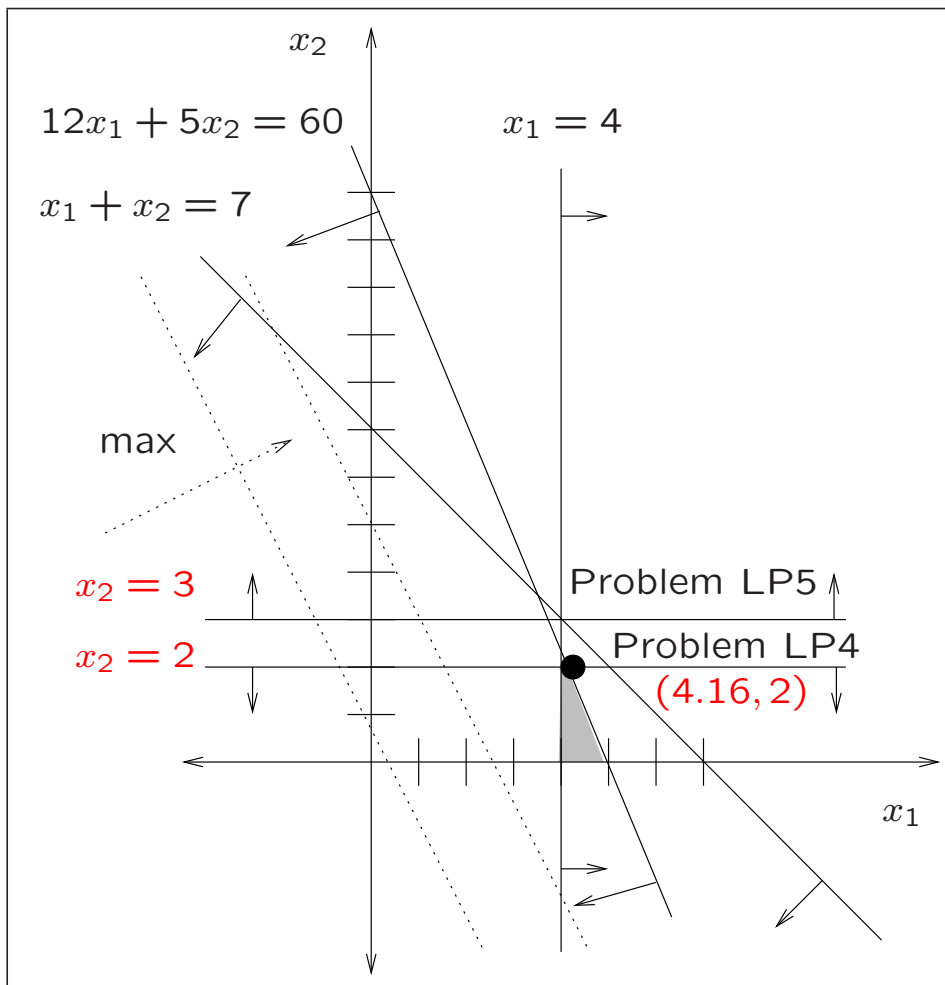$$x_1, x_2 \geq 0$$

Problem LP5

max $z = 80x_1 + 45x_2$

subject to

$$x_1 + x_2 \leq 7$$
$$12x_1 + 5x_2 \leq 60$$
$$x_1 \geq 4, \ x_2 \geq 3$$
$$x_1, x_2 \geq 0$$

Problem LP5 is pruned by infeasibility.

LP4: $\mathbf{x}_{LP4} = (4.166, 2)$, $z_{LP4} = 423.33 > z_{LB} = 420$.

Problem LP6

max $z = 80x_1 + 45x_2$

subject to

$$x_1 + x_2 \leq 7$$
$$12x_1 + 5x_2 \leq 60$$
$$x_1 \geq 4, \ x_2 \leq 2, \ x_1 \leq 4$$
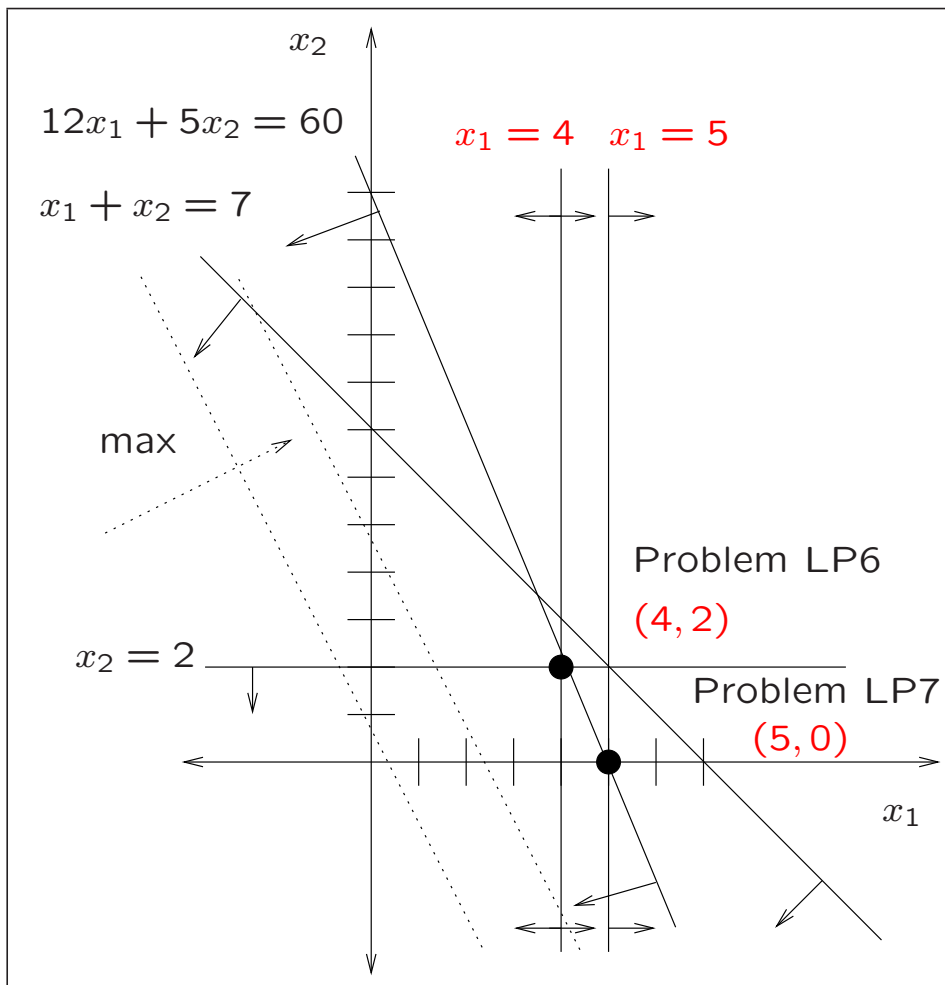$$x_1, x_2 \geq 0$$

Problem LP7

max $z = 80x_1 + 45x_2$

subject to

$$x_1 + x_2 \leq 7$$
$$12x_1 + 5x_2 \leq 60$$
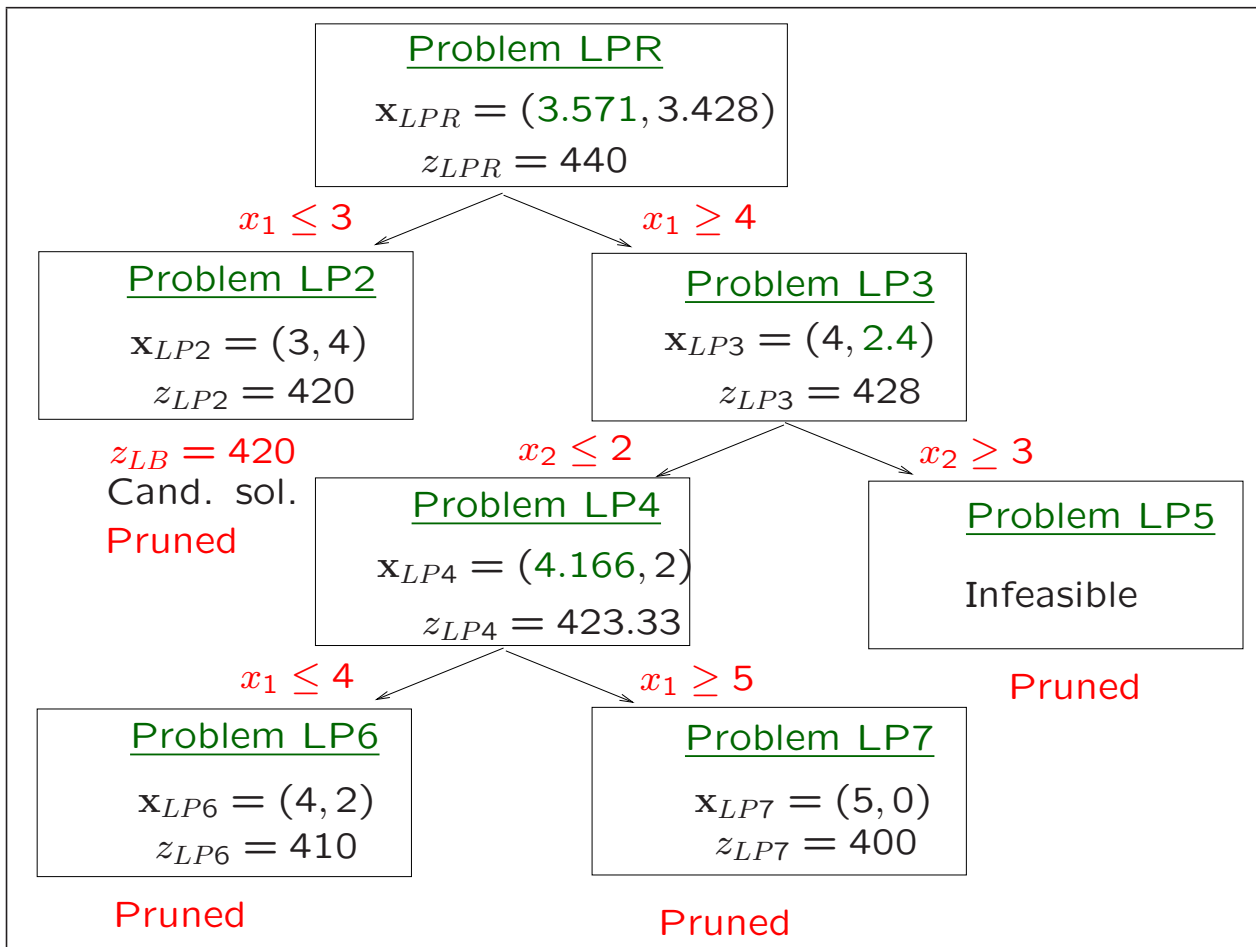$$x_1 \geq 4, x_2 \leq 2, x_1 \geq 5$$
$$x_1, x_2 \geq 0$$

$\mathbf{x}_{LP6} = (4, 2),$ $z_{LP6} = 410 < z_{LB} = 420 \rightarrow$ LP6 is pruned

$\mathbf{x}_{LP7} = (5, 0),$ $z_{LP7} = 400 < z_{LB} = 420 \rightarrow$ LP7 is pruned

The optimal solution to the IP problem:

$$\mathbf{x}^*_{IP} = \mathbf{x}_{LP2} = (x^*_1, x^*_2) = (3, 4)$$

$$z^*_{IP} = z_{LB} = 420$$

4. The branch and bound method

Definition 1 (LP relaxation) *Given an IP problem, the LP problem obtained by ignoring all integer constraints on variables is said to be its LP relaxation.*

IP Problem	LP relaxation: LPR
$\max \ z = \mathbf{c}^T \mathbf{x}$	$\max \ z \ = \mathbf{c}^T \mathbf{x}$
subject to	subject to
$\mathbf{Ax} \leq \mathbf{b}$	$\mathbf{Ax} \leq \mathbf{b}$
$\mathbf{x} \geq \mathbf{0}$ and integer	$\mathbf{x} \geq \mathbf{0}$

Definition 2 (Candidate solution) *Given an IP problem, an integer solution found throughout the solution process is said to be a candidate solution if it is the best integer solution found so far.*

Definition 3 (A pruned problem) *Throughout the solution process of an IP problem, the following three cases indicate that an LP problem can be pruned: (1) the LP problem is infeasible, (2) the optimal objective value of the LP problem is smaller than or equal to z_{LB}, (3) the LP problem has an integer optimal solution.*

The optimal objective value of an LP problem is an upper bound on the optimal objective value of the IP problem on that branch. We use the notation z_{UB}.

4.1 The branch and bound algorithm

The objective is to maximize.

* **Step 1. Initialization**. Solve the LP relaxation.

  - If the optimal solution to the LP relaxation satisfies the integer constraints, then it is an optimal solution to the IP problem. Stop.

  - Otherwise, set $z_{LB} = -\infty$ to initialize the lower bound on the optimal objective value of the IP problem.

* **Step 2. Branching**

  Select an LP problem among the LP problems that can be branched out. Choose a variable x_j which is integer-restricted in the IP problem but has a noninteger value in the optimal solution of the selected LP problem. Create two new LP problems adding the constraints $x_j \leq [x_j]$ and $x_j \geq [x_j] + 1$ to the LP problem.

* **Step 3. Bounding**

  Solve the two LP problems created in Step 2, and compute the objective value z_{UB} for each of them.

  Sensitivity analysis is commonly used and the dual simplex algorithm applied.

* Step 4. Pruning

An LP problem may be pruned and therefore eliminated from further consideration, in the following cases:

(1) Pruned by infeasibility. The problem is infeasible.

(2) Pruned by bound. $z_{UB} \leq z_{LB}$, that is, the optimal objective value of the LP problem is smaller than or equal to the lower bound.

(3) Pruned by optimality. The optimal solution is integer and $z_{UB} > z_{LB}$. Change the lower bound to the new value, $z_{LB} = z_{UB}$; the solution associated to the new lower bound is the new candidate solution.

If there are LP problems that can be branched out, then go to Step 2, and perform another iteration.

Otherwise, the candidate solution is the optimal solution to the IP problem.

If no candidate solution has been found, the IP problem is infeasible.

5. 0-1 integer programming

0-1 IP problems are linear problems where all the variables are binary.

Definition 4 (0-1 relaxation problem) *Given a 0-1 IP problem, the corresponding 0-1 relaxation problem is obtained by ignoring all constraints, except the ones which state that the variables are binary.*

Definition 5 (A partial solution) *Given a 0-1 IP problem, a solution where the values of some variables are unspecified is called a partial solution.*

Definition 6 (A completion of a partial solution) *Given a partial solution to a 0-1 IP problem, a completion of it is obtained by assigning a value to the variables with unspecified values in the partial solution.*

Before we apply the algorithm to solve a 0-1 IP problem, we need to make sure that the coefficients of the objective function satisfy the following:

$$0 \leq c_1 \leq c_2 \leq \cdots \leq c_n.$$

5.1 A 0-1 branch and bound algorithm

The objective is to maximize. The cost coefficients must satisfy: $0 \leq c_1 \leq c_2 \leq \cdots \leq c_n$.

* **Step 1. Initialization**

  If $\mathbf{x} = (1, \ldots, 1)$ satisfies the constraints of the original 0-1 IP problem, it is the optimal solution to the original 0-1 IP problem. Stop.

  Otherwise, if $\mathbf{x} = (0, 1, \ldots, 1)$ satisfies the constraints of the original 0-1 IP problem, it is the optimal solution to the original 0-1 IP problem. Stop.

  Otherwise, initialize the lower bound $z_{LB} = z(\mathbf{x})$, where $\mathbf{x} = (0, \ldots, 0)$.

  The upper bound associated to the 0-1 relaxation problem is $z_{UB} = z(\mathbf{x}_{UB})$, where $\mathbf{x}_{UB} = (0, 1, \ldots, 1)$.

  Assign the index $k = 1$ to the problem.

* **Step 2. Branching**

  Select a problem among the problems that can be branched out. Create two new problems by adding the constraints $x_k = 0$ and $x_k = 1$ to the problem selected.

* **Step 3. Bounding**

  For each of the two newly created problems, consider the completion $\mathbf{x}_{UB}$ which involves assigning the value 0 to the variable x_{k+1} and the value 1 to the rest of the unspecified variables. Compute the objective value z_{UB} for each of the completions. Assign the index $k = k + 1$ to the new problems.

* **Step 4. Pruning**

A problem may be pruned and therefore eliminated from further consideration, in the following cases:

(1) Pruned by bound. $z_{UB} \leq z_{LB}$.

(2) Pruned by optimality. $z_{UB} > z_{LB}$ and the completion $\mathbf{x}_{UB}$ satisfies the constraints of the original 0-1 IP problem. Change the lower bound to the new value, $z_{LB} = z_{UB}$; the solution $\mathbf{x}_{UB}$, which is associated to the new lower bound, is the new candidate solution.

(3) Pruned by infeasibility. None of the completions is a feasible solution to the original 0-1 IP problem.

If there are problems that can be branched out, then go to Step 2.

Otherwise, the candidate solution is the optimal solution to the 0-1 IP problem. Stop.