Chapter 6

Integer Programming

Integer programming (IP) deals with solving linear models in which some or all the variables are restricted to be integer. There are algorithms especially designed for IP problems which basically find the optimal solution by solving a sequence of linear programming (LP) problems.

The simplex algorithm studied in Chapter 2 is based on the fact that the feasible region (the set of feasible solutions) of an LP problem is convex. This property plays a key role in the solution of linear models. In fact, the number of extreme points of a convex set of solutions is finite, and we have shown that the optimal solution is obtained in an extreme point. Therefore, even though the number of solutions is reduced when variables are restricted to be integer, IP problems are usually much more difficult to solve than LP problems because the set of feasible solutions is no longer convex.

According to the nature of the variables, we can distinguish 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).

6.1 Some applications of integer programming

This section presents some illustrative examples of typical integer programming problems (IP problems) and binary programming problems (0-1 IP problems).

Example 1. The number of employees needed in a post office varies depending on the day of the week, as shown in the table:

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. It becomes necessary to organize groups of employees to work in different shifts, so that the number of employees required is satisfied every day of the week. The objective is to employ the minimum number of workers. We define the following decision variables:

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

We can make sure that the number of employees required every day of the week is satisfied by introducing a constraint in the model. Each of the constraints requires to have the necessary number of employees each day. We obtain the following IP model:

$$
\begin{aligned}
\min\ z &= x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \\
\text{subject to} & \\
x_1 + x_4 + x_5 + x_6 + x_7 &\geq 15 \\
x_1 + x_2 + x_5 + x_6 + x_7 &\geq 13 \\
x_1 + x_2 + x_3 + x_6 + x_7 &\geq 15 \\
x_1 + x_2 + x_3 + x_4 + x_7 &\geq 18 \\
x_1 + x_2 + x_3 + x_4 + x_5 &\geq 14 \\
x_2 + x_3 + x_4 + x_5 + x_6 &\geq 16 \\
x_3 + x_4 + x_5 + x_6 + x_7 &\geq 10 \\
x_1, x_2, x_3, x_4, x_5, x_6, x_7 &\geq 0 \text{ and integer}
\end{aligned}
$$

Example 2. A knapsack problem. Let us suppose that we want to put four items in a knapsack that can hold up to 12 kg. The weight and the value associated with each of the items are listed below:

	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, $j = 1, 2, 3, 4$, as follows:

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

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

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

We may also want to take into account other constraints in the problem, such as the volume of the items.

Example 3. There are 6 cities in a district, and there is a project to communicate them by the construction of train stations in some of the cities. A decision has to be made about where to build the train stations. The final solution has to ensure that citizens of all cities can reach a train station in 30 minutes at most. The objective is to build the minimum number of train stations. The table below shows how long it takes to go from one city to another:

	1	2	3	4	5	6
1	0	35	20	40	30	60
2	35	0	45	35	20	70
3	20	45	0	15	55	20
4	40	35	15	0	65	35
5	30	20	55	65	0	40
6	60	70	20	35	40	0

We define a binary variable for each city j, $j = 1, \ldots, 6$:

$$x_j = \begin{cases} 1 & \text{if a train station is built in city } j \\ 0 & \text{otherwise} \end{cases}$$

The 0-1 IP model is:

$$\begin{aligned} \min\ z &= x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\ &\text{subject to} \\ x_1 + x_3 + x_5 &\geq 1 \\ x_2 + x_5 &\geq 1 \\ x_1 + x_3 + x_4 + x_6 &\geq 1 \\ x_3 + x_4 &\geq 1 \\ x_1 + x_2 + x_5 &\geq 1 \\ x_3 + x_6 &\geq 1 \\ x_1, x_2, x_3, x_4, x_5, x_6 &= 0 \text{ or } 1 \end{aligned}$$

Each constraint refers to a city, and ensures that at least one train station will be located no further than a 30 minute drive from each city.

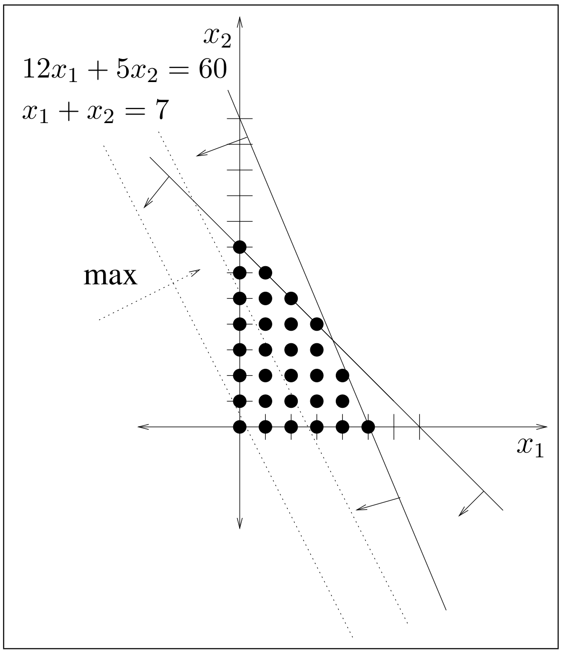
6.2 Solving integer programming problems

In this section we illustrate by means of an example the difficulties found while solving an IP problem.

Consider the following IP problem:

$$\begin{aligned} \max\ z = 80x_1 + 45x_2 \\ \text{subject to} \\ x_1 + x_2 \le 7 \\ 12x_1 + 5x_2 \le 60 \\ x_1, x_2 \ge 0 \text{ and integer} \end{aligned}$$

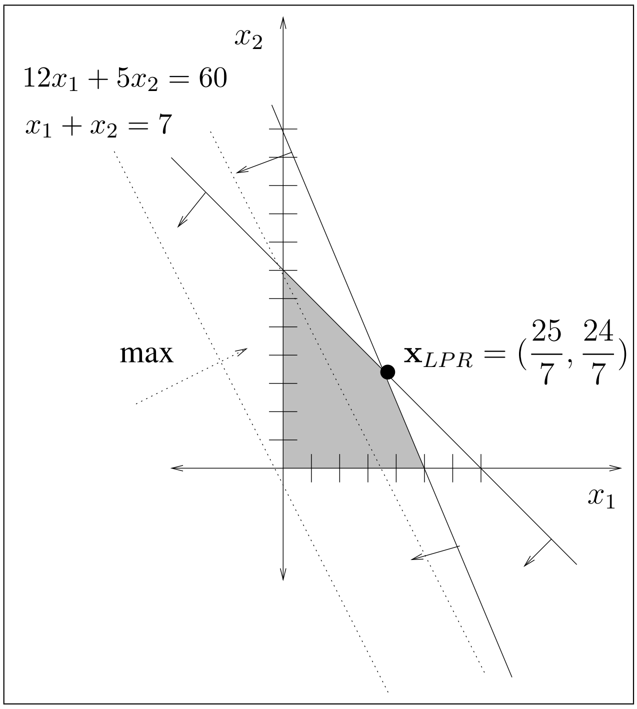
The graphical representation of the problem shows the set of solutions:



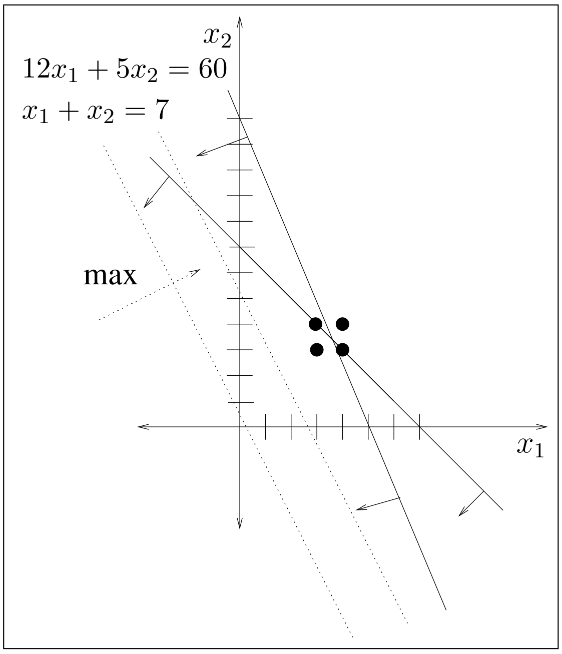
We can see that the feasible region of the IP problem is not a convex set. Note that there is a finite number of points in the feasible region, and hence, it is possible to find the optimal solution by computing the objective value z for each of the solutions in the feasible region, and comparing them among each other. However, this method is not efficient for problems with a large number of variables, since the number of feasible points becomes extremely large.

In fact, a higher computational effort is required to solve an IP problem than to solve the LP problem obtained by ignoring all integer constraints on variables, even though the number of feasible solutions to the IP problem is smaller. This is the case because contrary to LP problems, which have a convex set of feasible solutions, the feasible region of IP problems is not convex. Remind that the theory developed in Chapter 2 is applicable whenever the set of solutions is convex.

Another approach to solve an IP problem suggests to ignore the integer constraints, solve the resulting LP problem by applying the simplex algorithm and finally round off the noninteger values to integers. The LP problem obtained by ignoring the integer constraints is usually called its *LP relaxation*, denoted by LPR from now on. The following is the graphical solution of the given IP problem's LP relaxation.



The optimal solution to the LP relaxation is $\mathbf{x}_{LPR} = (\frac{25}{7}, \frac{24}{7}) = (3.571, 3.428)$, and the optimal objective value is $z_{LPR} = 440$. However, it is not an optimal solution to the IP problem, because it is not feasible; it does not satisfy the integer constraints. Rounding off each integer variable to the nearest integer value, we obtain the following four nearest points: $(3, 3)$, $(3, 4)$, $(4, 3)$, $(4, 4)$. By computing the objective value z for each of them, we conclude that point $(4, 4)$ gives the maximum value to z. Unfortunately, it is not feasible because it is not contained in the set of solutions of the IP problem.



This method is not appropriate to solve IP problems, because there is no guarantee that the rounded solution will be optimal, or even feasible, for the IP problem. Moreover, for IP problems with a large number of integer variables the difficulties increase.

Because of these difficulties, better approaches to deal with IP problems have been devised. Next, we present a very popular technique called the branch and bound method.

6.3 The graphical solution of integer programming problems

The basic idea of the branch and bound algorithm is the following. First, the LP relaxation of the IP problem is solved. If its optimal solution does not satisfy the integer requirements, two additional LP problems are created by subdividing the set of solutions of the LP relaxation. This partitioning is made in such a way that a subset of noninteger solutions that contains the optimal solution to the LP relaxation is excluded from the set of solutions, and gives rise to the concept of branching in the branch and bound algorithm. Afterwards, the two new LP problems are solved.

In this section, we illustrate the branch and bound algorithm by applying it to the IP problem shown on page 209. A sequence of LP relaxation problems are used to solve the IP problem, and their graphical solution represents the sets of solutions very appropriately. Let us consider the IP problem and its LP relaxation.

IP problem	LP relaxation: LPR
$\max\ z = 80x_1 + 45x_2$	$\max\ z = 80x_1 + 45x_2$
subject to	subject to
$x_1 + x_2 \leq 7$	$x_1 + x_2 \leq 7$
$12x_1 + 5x_2 \leq 60$	$12x_1 + 5x_2 \leq 60$
$x_1, x_2 \geq 0$ and integer	$x_1, x_2 \geq 0$

From the graphical solution on page 210, we know that the optimal solution to the LP relaxation is $\mathbf{x}_{LPR} = (3.571, 3.428)$, with the optimal objective value $z_{LPR} = 440$. This solution does not give integer values to the variables. We will now see that the optimal solution to the IP problem may be found throught the solution of a sequence of LP relaxation problems. To do so, we start the branching process dividing the set of solutions of the LP relaxation into two, and excluding a subset of noninteger solutions that contains $\mathbf{x}_{LPR}$.

The way to proceed is as follows: we select an integer defined variable that takes on a fractional value in the optimal solution to the LP relaxation. In this example, both x_1 and x_2 may be selected. Selecting x_1 arbitrarily, with an optimal value of 3.571, we may exclude the region $3 < x_1 < 4$ from the feasible region of the LP relaxation, because it contains no integer values for x_1. Note that every point in the feasible region of the IP problem must have either $x_1 \leq 3$ or $x_1 \geq 4$. Therefore, we partition the feasible region of the LP relaxation by branching on x_1, and create the following two additional LP problems:

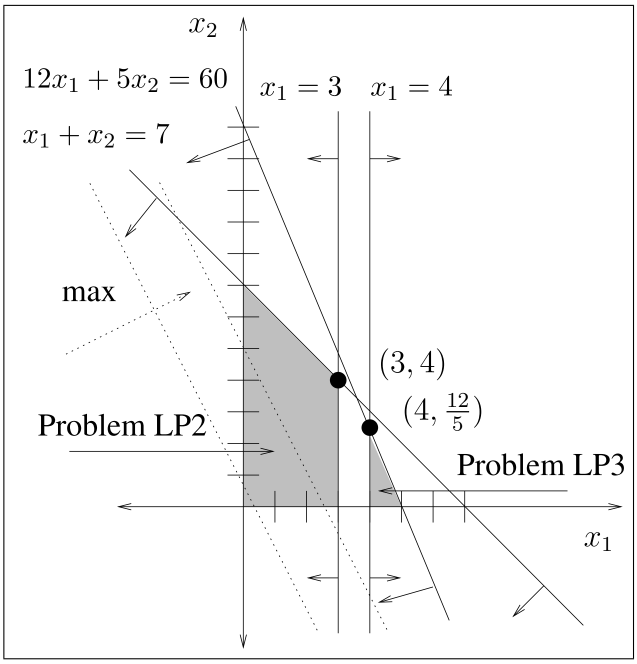
Problem LP2	Problem LP3
$\max\ z = 80x_1 + 45x_2$	$\max\ z = 80x_1 + 45x_2$
subject to	subject to
$x_1 + x_2 \leq 7$	$x_1 + x_2 \leq 7$
$12x_1 + 5x_2 \leq 60$	$12x_1 + 5x_2 \leq 60$
$x_1 \leq 3$	$x_1 \geq 4$
$x_1, x_2 \geq 0$	$x_1, x_2 \geq 0$

The graphical solution of the two LP problems just created, LP2 and LP3, can be seen on page 214. The two shaded areas correspond to the feasible regions of problems LP2 and LP3. Note that the region $3 < x_1 < 4$ of the LP relaxation feasible region has been excluded. The optimal solutions are:

- Problem LP2: Optimal solution $\mathbf{x}_{LP2} = (3, 4)$ with $z_{LP2} = 420$.

- Problem LP3: Optimal solution $\mathbf{x}_{LP3} = (4, \frac{12}{5})$ with $z_{LP3} = 428$.

The optimal solution to problem LP2 satisfies the integer requirements for x_1 and x_2. Thus, LP2 is said to be *fathomed* or *pruned*, which means that no further branching is required for LP2. The optimal solution to LP2, $\mathbf{x}_{LP2} = (3, 4)$, is called a *candidate solution*, which means that it will be an optimal solution for the IP problem, if a better feasible solution is not found. The optimal objective value $z_{LP2} = 420$ for the candidate solution is a *lower bound* on the optimal objective value of the IP problem: $z_{LB} = 420$.

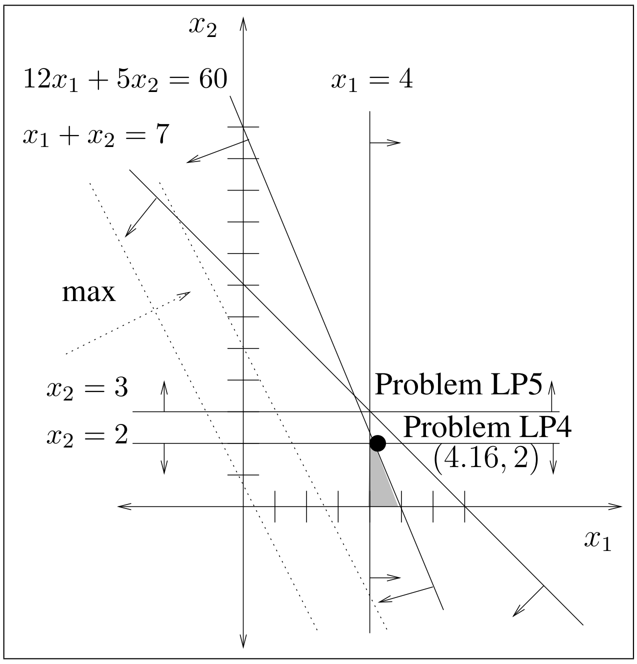


We now examine problem LP3. The optimal solution to problem LP3 is not feasible for the IP problem, because variable $x_2 = \frac{12}{5} = 2.4$ takes on a fractional value. Since $z_{LP3} = 428 > z_{LB}$, problem LP3 is examined further because branching on LP3 may yield a better feasible integer solution than the candidate solution; a solution with $z > 420$.

The fractional value of x_2 leads to the two branches $x_2 \leq 2$ and $x_2 \geq 3$. Branching on LP3, the two additional problems LP4 and LP5 are generated. The integer-free region $2 < x_2 < 3$ is excluded from the feasible region of problem LP3.

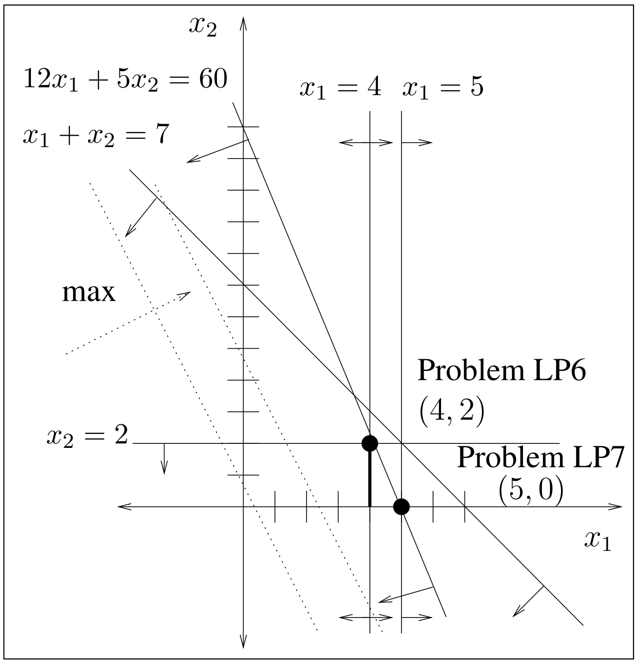
Problem LP4	Problem LP5
$\max\ z = 80x_1 + 45x_2$	$\max\ z = 80x_1 + 45x_2$
subject to	subject to
$x_1 + x_2 \leq 7$	$x_1 + x_2 \leq 7$
$12x_1 + 5x_2 \leq 60$	$12x_1 + 5x_2 \leq 60$
$x_1 \geq 4,\ x_2 \leq 2$	$x_1 \geq 4,\ x_2 \geq 3$
$x_1, x_2 \geq 0$	$x_1, x_2 \geq 0$

From the graphical solution of the two newly created problems, we see that problem LP5 is infeasible. It cannot yield the optimal solution to the IP problem, and thus, problem LP5 is pruned; no further branching is required.



The optimal solution to problem LP4 is $\mathbf{x}_{LP4} = (\frac{25}{6}, 2) = (4.166, 2)$, which is not feasible for the IP problem, because $x_1 = 4.166$ is noninteger. Since $z_{LP4} = \frac{1270}{3} = 423.33$ and compared with the lower bound $z_{LP4} > z_{LB} = 420$ holds, problem LP4 is examined further, because branching on LP4 may yield a better feasible integer solution than the candidate solution. The noninteger value of x_1 leads to the two branches $x_1 \leq 4$ and $x_1 \geq 5$. Branching on LP4, the two additional problems LP6 and LP7 are generated.

Problem LP6	Problem LP7
$\max\ z = 80x_1 + 45x_2$	$\max\ z = 80x_1 + 45x_2$
subject to	subject to
$x_1 + x_2 \leq 7$	$x_1 + x_2 \leq 7$
$12x_1 + 5x_2 \leq 60$	$12x_1 + 5x_2 \leq 60$
$x_1 \geq 4,\ x_2 \leq 2,\ x_1 \leq 4$	$x_1 \geq 4, x_2 \leq 2, x_1 \geq 5$
$x_1, x_2 \geq 0$	$x_1, x_2 \geq 0$



The feasible region of problem LP6 is a line segment, and its optimal solution is $\mathbf{x}_{LP6} = (4, 2)$ with $z_{LP6} = 410$. Since $z_{LP6} < z_{LB} = 420$, the problem is pruned.

The feasible region of problem LP7 contains just one point, which consequently is its optimal solution: $\mathbf{x}_{LP7} = (5, 0)$. The optimal objective value is $z_{LP7} = 400$, which is lower than the lower bound, $z_{LP7} < z_{LB} = 420$. Hence, the problem is pruned.

At this point, no further branching is required. Since there are no remaining unsolved problems, the optimal solution to the IP problem is the candidate solution obtained from problem LP2, that is, the one associated with the lower bound:

$$\mathbf{x}^*_{IP} = \mathbf{x}_{LP2} = (x_1^*, x_2^*) = (3,4) \quad \text{and} \quad z^*_{IP} = z_{LB} = 420.$$

The entire solution sequence is summarized in a diagram (see Figure 6.1). Note that we completed the procedure by solving a total of seven LP problems. Also note that the optimal objective value computed for each LP problem is an upper bound on the optimal objective value of the IP problem on that branch.

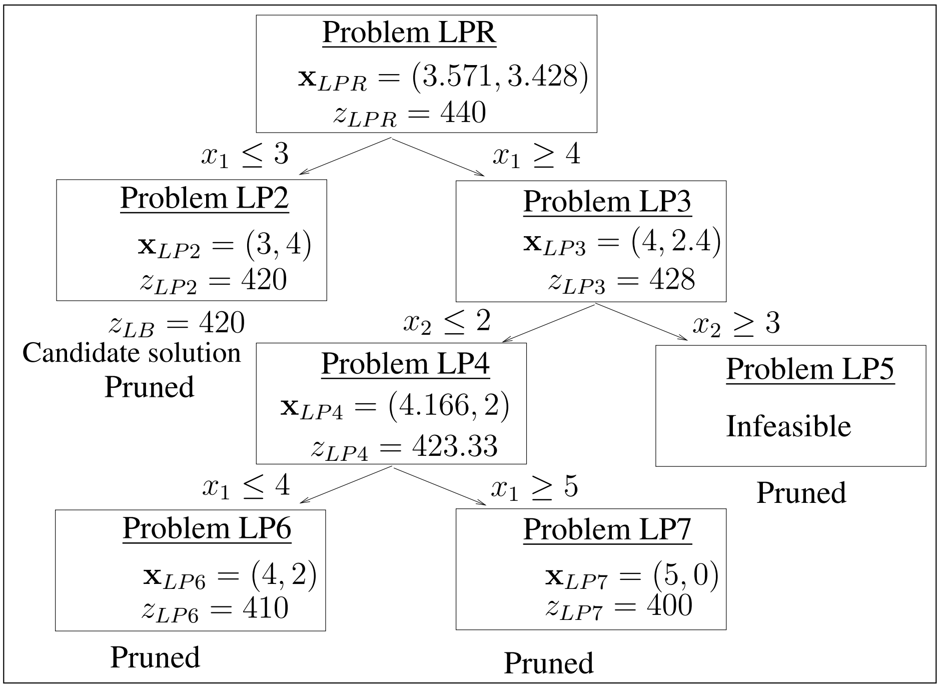


Figure 6.1: Diagram of the entire solution sequence of the example.

6.4 The branch and bound method

In the previous section, we used the branch and bound algorithm and solved graphically a sequence of LP problems to find the optimal solution to an IP problem. Throughout the solution process, we used the following concepts: the *LP relaxation* of an IP problem, a *candidate solution* and a *fathomed* or *pruned* LP problem.

Definition 6.4.1 (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
$\mathbf{A}\mathbf{x} \leq \mathbf{b}$	$\mathbf{A}\mathbf{x} \leq \mathbf{b}$
$\mathbf{x} \geq \mathbf{0}$ and integer	$\mathbf{x} \geq \mathbf{0}$

The LP relaxation has less constraints than the IP problem, because all integer constraints on variables are ignored. Therefore, the set of all feasible solutions to the LP relaxation includes all the feasible solutions to the IP problem. Consequently, the following holds:

$$z^*_{LPR} \geq z^*_{IP}.$$

Definition 6.4.2 (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.*

A candidate solution will become an optimal solution to the IP problem, if at the end of the branch and bound algorithm a better integer solution is not found. The optimal objective value z_{LB} for the candidate solution is a *lower bound* on the optimal objective value of the IP problem. In fact, it is the largest objective value computed for a solution which meets all the integer constraints. Throughout the solution process of the IP problem, if the optimal objective value of an LP problem is smaller than or equal to z_{LB}, then the LP problem is *pruned* and it will not be examined further, because branching on it will not yield a better solution to the IP problem.

Definition 6.4.3 (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.*

For instance, problems LP2, LP5, LP6 and LP7 are pruned problems (see Figure 6.1).

As it was previously said, 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} to denote the *upper bound* that the optimal objective value of each LP problem establishes throughout the solution process of an IP problem.

6.4.1 The branch and bound algorithm

Let us assume we have a maximization IP problem. The branch and bound algorithm can be summarized in the following steps:

* **Step 1. Initialization**

  Solve the LP relaxation associated with the IP problem to be solved.

  – 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[1] $x_j \leq [x_j]$ and $x_j \geq [x_j] + 1$ to the LP problem.

* **Step 3. Bounding**

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

* **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.

[1] $[x_j]$ represents the greatest integer less than or equal to x_j

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

(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 with 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.

Even though a high computational effort is required to find the optimal solution to an IP problem by applying the branch and bound algorithm, it is the most popular algorithm used to solve both mixed and pure IP problems.

Note that Step 2 is quite flexible, because it does not specify neither how to select an LP problem to be branched out nor how to choose a branching variable x_j, if there are several choices. Several rules have been designed to avoid arbitrary choices and guide the search of an optimal solution to the IP problem. In fact, experience has shown that the way such decisions are made has an important effect on the computational efficiency of the branch and bound algorithm. A commonly used rule to select an LP problem to be branched out is the *best bound rule*, which suggests that the LP problem with the largest upper bound z_{UB} should be selected. Some rules have also been designed to choose a branching variable, but unfortunately, they are quite complex. In the following example, we choose the branching variable arbitrarily.

Example. We apply the branch and bound algorithm to find the optimal solution to the IP problem shown on page 209.

First iteration

Step 1. Initialization. Solve the LP relaxation associated with the IP problem. The optimal tableau is:

	x_1	x_2	x_3	x_4	
	0	0	20	5	440
$\mathbf{a}_2$	0	1	$\frac{12}{7}$	$-\frac{1}{7}$	$\frac{24}{7}$
$\mathbf{a}_1$	1	0	$-\frac{5}{7}$	$\frac{1}{7}$	$\frac{25}{7}$

Set $z_{LB} = -\infty$ to initialize the lower bound.

Step 2. Branching. The optimal solution to the LP relaxation is not integer. We choose the branching variable, x_1 for instance, and create two new problems: problem LP2 and problem LP3 (see page 212).

Step 3. Bounding. We solve the two LP problems created in Step 2 using the sensitivity analysis and the dual simplex algorithm.

- **Solving problem LP2.** We add a slack variable to the constraint $x_1 \leq 3$ and include it into the optimal tableau associated with the LP relaxation LPR. This yields the following tableau:

	x_1	x_2	x_3	x_4	x_5	
	0	0	20	5	0	440
$\mathbf{a}_2$	0	1	$\frac{12}{7}$	$-\frac{1}{7}$	0	$\frac{24}{7}$
$\mathbf{a}_1$	1	0	$-\frac{5}{7}$	$\frac{1}{7}$	0	$\frac{25}{7}$
$\mathbf{a}_5$	1	0	0	0	1	3

We need to use elementary operations to write the third row in terms of the new basis $\mathbf{B} = (\mathbf{a}_2\ \mathbf{a}_1\ \mathbf{a}_5)$. We operate like this: row $3 -$ row 2.

	x_1	x_2	x_3	x_4	x_5	
	0	0	20	5	0	440
$\mathbf{a}_2$	0	1	$\frac{12}{7}$	$-\frac{1}{7}$	0	$\frac{24}{7}$
$\mathbf{a}_1$	1	0	$-\frac{5}{7}$	$\frac{1}{7}$	0	$\frac{25}{7}$
$\mathbf{a}_5$	0	0	$\frac{5}{7}$	$-\frac{1}{7}$	1	$-\frac{4}{7}$

The tableau is not primal feasible; the dual simplex algorithm will be used to find the optimal tableau for the problem LP2.

	x_1	x_2	x_3	x_4	x_5	
	0	0	45	0	35	420
$\mathbf{a}_2$	0	1	1	0	-1	4
$\mathbf{a}_1$	1	0	0	0	1	3
$\mathbf{a}_4$	0	0	-5	1	-7	4

- **Solving problem LP3.** We multiply constraint $x_1 \geq 4$ by -1 to include it in the optimal tableau associated with the problem LPR, $-x_1 \leq -4$, and add the slack variable x_5. This yields the following tableau:

	x_1	x_2	x_3	x_4	x_5	
	0	0	20	5	0	440
$\mathbf{a}_2$	0	1	$\frac{12}{7}$	$-\frac{1}{7}$	0	$\frac{24}{7}$
$\mathbf{a}_1$	1	0	$-\frac{5}{7}$	$\frac{1}{7}$	0	$\frac{25}{7}$
$\mathbf{a}_5$	-1	0	0	0	1	-4

To adjust the third row, we operate like this: row $3 +$ row 2.

	x_1	x_2	x_3	x_4	x_5	
	0	0	20	5	0	440
$\mathbf{a}_2$	0	1	$\frac{12}{7}$	$-\frac{1}{7}$	0	$\frac{24}{7}$
$\mathbf{a}_1$	1	0	$-\frac{5}{7}$	$\frac{1}{7}$	0	$\frac{25}{7}$
$\mathbf{a}_5$	0	0	$-\frac{5}{7}$	$\frac{1}{7}$	1	$-\frac{3}{7}$

The tableau is not primal feasible; the dual simplex algorithm will be used to find the optimal tableau for the problem LP3.

	x_1	x_2	x_3	x_4	x_5	
	0	0	0	9	28	428
$\mathbf{a}_2$	0	1	0	$\frac{1}{5}$	$\frac{12}{5}$	$\frac{12}{5}$
$\mathbf{a}_1$	1	0	0	0	-1	4
$\mathbf{a}_3$	0	0	1	$-\frac{1}{5}$	$-\frac{7}{5}$	$\frac{3}{5}$

Thereby, Problems LP2 and LP3 have been solved (Figure 6.1 on page 217 shows the optimal solutions).

Step 4. Pruning.

Problem LP2 is pruned by optimality, because $z_{UB} = 420 > z_{LB}$ holds and the solution is integer: $x_1 = 3$ and $x_2 = 4$. Since it is the first integer solution found so far, it becomes the candidate solution, and we change the lower bound to the new value: $z_{LB} = z_{UB} = 420$.

Problem LP3 is not pruned, because none of the conditions of Step 4 holds.

Since there are LP problems that can be branched out, we proceed with a new iteration of the algorithm.

Second iteration

Step 2. Branching. We select problem LP3 to be branched out. We choose variable x_2, and create two new LP problems: problem LP4, by adding the constraint $x_2 \leq 2$ to problem LP3, and problem LP5, by adding the constraint $x_2 \geq 3$ (see page 214).

Step 3. Bounding. As in the first iteration, we solve the two LP problems just created using the sensitivity analysis and the dual simplex algorithm. This time, we use the optimal tableau associated with problem LP3 to solve problems LP4 and LP5 (see the solutions on Figure 6.1, page 217).

Step 4. Pruning. Problem LP5 is pruned by infeasibility. The optimal solution to problem LP4 is not integer, and $z_{UB} = 423.33 > 420 = z_{LB}$. Thus, problem LP4 is not pruned; it is examined further, because branching on LP4 may yield a better feasible integer solution than the candidate solution. We proceed with a new iteration of the algorithm at Step 2.

Third iteration

Step 2. Branching. We select problem LP4 to be branched out. We choose variable x_1, and create two new LP problems: problem LP6 and problem LP7 (see models on page 215).

Step 3. Bounding. We solve the two LP problems just created. This time, we start from the optimal tableau associated with problem LP4 to solve problems LP6 and LP7 (see the solutions on Figure 6.1, page 217).

Step 4. Pruning.

Problem LP6 is pruned by bound, because $z_{UB} = 410 < 420 = z_{LB}$ holds.

Problem LP7 is also pruned by bound, since $z_{UB} = 400 < 420 = z_{LB}$ holds.

No problem remains to be branched out. Therefore, the candidate solution is the optimal solution to the IP problem.

$$x_1^* = 3, \;\; x_2^* = 4, \;\; z_{IP}^* = z_{LB} = 420.$$

□

6.5 0-1 integer programming

In practice, there are problems where all the variables are binary, and for the solution of which different algorithms have been proposed. In this section, we present one which has basically the same structure as the branch and bound algorithm introduced earlier.

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:

$$0 \leq c_1 \leq c_2 \leq \cdots \leq c_n \tag{6.1}$$

It is always possible to rewrite the 0-1 IP model and obtain a form that satisfies the condition (6.1). Let us see how to proceed by means of an example.

Example. Consider the following 0-1 IP model:

$$\begin{aligned} \max\ z = 6x_1 - 4x_2 & \\ \text{subject to} & \\ 3x_1 + 2x_2 &\leq 10 \\ -x_1 + x_2 &\leq 17 \\ x_1, x_2 &= 0 \text{ or } 1 \end{aligned}$$

Cost coefficients in the objective function do not satisfy the condition (6.1). To obtain the required form, we proceed like this: we consider the absolute value of the cost coefficients and choose the minimum: $min\{|c_1|, |c_2|\} = min\{6, 4\} = 4$. We replace the binary variable x_2 by $1 - y_1$, because c_2 is negative. If it was positive, it should be replaced by y_1. The next smaller is $|c_1| = 6$; since it is positive, we replace x_1 by y_2. Both y_1 and y_2 are also binary variables.

All the variables can now be reordered as needed to place the cost coefficients in ascending order, so that the condition (6.1) holds.

$$
\begin{aligned}
\max\ z = 4y_1 + 6y_2 - 4 \\
\text{subject to} \quad\quad \\
-2y_1 + 3y_2 \leq 8 \\
-y_1 - y_2 \leq 16 \\
y_2, y_2 = 0 \text{ or } 1
\end{aligned}
$$

□

Definition 6.5.1 (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 6.5.2 (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 6.5.3 (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.*

Example. Consider the following 0-1 IP problem:

$$
\begin{aligned}
\max\ z = x_1 + 2x_2 + 4x_3 \\
\text{subject to} \quad\quad \\
x_1 + x_2 + 2x_3 \leq 4 \\
3x_1 + x_2 + 2x_3 \leq 5 \\
x_1, x_2, x_3 = 0 \text{ or } 1
\end{aligned}
$$

The corresponding 0-1 relaxation problem is the following:

$$\max\ z = x_1 + 2x_2 + 4x_3$$
$$\text{subject to}$$
$$x_1, x_2, x_3 = 0 \text{ or } 1$$

For instance, $\mathbf{x} = (1, 1, -)$ is a partial solution to the 0-1 relaxation problem. There are two possible completions of that partial solution: $(1, 1, 0)$ and $(1, 1, 1)$. $\mathbf{x} = (0, -, -)$ is also a partial solution to the 0-1 relaxation problem, which has the following four possible completions: $(0, 1, 1)$, $(0, 1, 0)$, $(0, 0, 1)$ and $(0, 0, 0)$.

It is easy to solve the 0-1 relaxation problem because all the constraints of the original 0-1 IP problem have been ignored. Moreover, since all the cost coefficients are positive, it is clear that the optimal solution to the 0-1 relaxation problem is $\mathbf{x}^* = (1, 1, 1)$. If we check and see that it is not feasible for the original 0-1 IP problem, that is, if it does not satisfy its constraints, then we can check the next best solution to the 0-1 relaxation problem; we know how to compute it, because the cost coefficients are placed in ascending order in the objective function. Thus, we check the solution $\mathbf{x} = (0, 1, 1)$, the one that makes the objective value largest, once $\mathbf{x}^* = (1, 1, 1)$ has been discarded. In fact, we can order the solutions to the 0-1 relaxation problem from the best to the worst, and check orderly from best to worst whether the solutions are feasible for the original 0-1 IP problem. This implicit enumeration stops when a feasible solution is found.

□

The idea at the basis of the 0-1 branch and bound algorithm presented in the next section is precisely to start at the optimal solution to the 0-1 relaxation problem, and check whether it satisfies the constraints of the original 0-1 IP problem. The aim of branching is to look for the optimal solution to the 0-1 IP problem by solving a sequence of 0-1 relaxation problems.

6.5.1 A 0-1 branch and bound algorithm

The algorithm presented in this section was designed to solve 0-1 IP problems where the objective is to maximize. The cost coefficients in the objective function must satisfy the condition $0 \leq c_1 \leq c_2 \leq \cdots \leq c_n$ before the algorithm is applied.

* **Step 1. Initialization**

  Consider the optimal solution to the 0-1 relaxation problem, $\mathbf{x} = (1, \ldots, 1)$, and check whether it satisfies the constraints of the original 0-1 IP problem.

If that is the case, $(1, \ldots, 1)$ is the optimal solution to the original 0-1 IP problem. Stop.

Otherwise, check whether the solution $\mathbf{x} = (0, 1, \ldots, 1)$ satisfies the constraints of the original 0-1 IP problem. If that is the case, $(0, 1, \ldots, 1)$ 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 with 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 with 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.

Example. We now illustrate the 0-1 branch and bound algorithm by applying it to the knapsack problem of the example on page 207.

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

In order to place the cost coefficients of the objective function in ascending order, we make these changes in variables: $x_4 = y_1$, $x_3 = y_2$, $x_1 = y_3$ and $x_2 = y_4$. The changes give us the following 0-1 IP problem (see also its corresponding 0-1 relaxation problem):

<u>0-1 IP problem</u>

$$\begin{aligned} \max\ z = 10y_1 + 12y_2 + 15y_3 + 25y_4 \\ \text{subject to} \qquad\qquad \\ 5y_1 + 5y_2 + 3y_3 + 6y_4 \leq 12 \\ y_1, y_2, y_3, y_4 = 0 \text{ or } 1 \end{aligned}$$

<u>0-1 Relaxation Problem: 0-1 RP</u>

$$\begin{aligned} \max\ z = 10y_1 + 12y_2 + 15y_3 + 25y_4 \\ \text{subject to} \qquad\qquad \\ y_1, y_2, y_3, y_4 = 0 \text{ or } 1 \end{aligned}$$

The entire solution sequence is summarized in a diagram (see Figure 6.2 on page 232). All the 0-1 relaxation problems created while looking for the optimal solution to the original 0-1 IP problem using the 0-1 branch and bound algorithm, a partial solution to each of them, a completion and its corresponding upper bound are shown in the diagram.

First iteration

Step 1. Initialization.

The optimal solution to the 0-1 relaxation problem, $(1, 1, 1, 1)$, is not feasible for the original 0-1 IP problem because it does not satisfy its constraint.

The next best solution to the 0-1 relaxation problem is $(0, 1, 1, 1)$, but it is also infeasible for the original 0-1 IP problem because, likewise, it does not satisfy its constraint. The objective value at this solution is $z_{UB} = 52$.

We assign the index $k = 1$ to the problem, and initialize the lower bound, $z_{LB} = 0$.

Step 2. Branching.

We create two new problems by adding the constraints $y_1 = 0$ and $y_1 = 1$ to the 0-1 relaxation problem. This yields the new problems RP2 and RP3, respectively.

Problem RP2	Problem RP3
$\max\ z = 10y_1 + 12y_2 + 15y_3 + 25y_4$	$\max\ z = 10y_1 + 12y_2 + 15y_3 + 25y_4$
subject to	subject to
$y_1 = 0$	$y_1 = 1$
$y_2, y_3, y_4 = 0$ or 1	$y_2, y_3, y_4 = 0$ or 1

Step 3. Bounding.

The completion $\mathbf{y}_{UB} = (0, 0, 1, 1)$ for problem RP2 provides an upper bound for the 0-1 IP problem at its corresponding resolution branch: $z_{UB} = 40$.

The completion $\mathbf{y}_{UB} = (1, 0, 1, 1)$ for problem RP3 provides an upper bound for the 0-1 IP problem at its corresponding resolution branch: $z_{UB} = 50$.

We assign the index $k = 2$ to these two problems.

Step 4. Pruning.

The completion $\mathbf{y}_{UB} = (0, 0, 1, 1)$ for problem RP2 satisfies the constraint of the original 0-1 IP problem. Moreover, since $z_{UB} = 40 > 0 = z_{LB}$ holds, the completion becomes the candidate solution. Problem RP2 is pruned by optimality, and the lower bound changes to the new value, $z_{LB} = 40$.

The completion $\mathbf{y}_{UB} = (1, 0, 1, 1)$ for problem RP3, however, violates the constraint of the original 0-1 IP problem. Moreover, we can check that problem RP3 is not infeasible, because there exists at least one feasible completion for it; one that satisfies the constraint of the original 0-1 IP problem, say $\mathbf{y} = (1, 0, 0, 0)$, for instance. Since $z_{UB} = 50 > z_{LB}$, none of the conditions for pruning holds. Therefore, RP3 is not pruned, and we proceed with a new iteration of the algorithm.

Second iteration.

We consider problem RP3. By branching on y_2, we create two additional problems: adding the constraint $y_2 = 0$ we get problem RP4, and adding the constraint $y_2 = 1$ we get problem RP5.

Problem RP4

$$\max\ z = 10y_1 + 12y_2 + 15y_3 + 25y_4$$

subject to

$$y_1 = 1$$
$$y_2 = 0$$
$$y_3, y_4 = 0 \text{ or } 1$$

Problem RP5

$$\max\ z = 10y_1 + 12y_2 + 15y_3 + 25y_4$$

subject to

$$y_1 = 1$$
$$y_2 = 1$$
$$y_3, y_4 = 0 \text{ or } 1$$

We compute the completions and the objective values for problems RP4 and RP5 as specified in the algorithm.

For problem RP4, we have the completion $\mathbf{y}_{UB} = (1, 0, 0, 1)$ with $z_{UB} = 35$. Problem RP4 is pruned by bound, because $z_{UB} < z_{LB} = 40$.

For problem RP5, we have the completion $\mathbf{y}_{UB} = (1, 1, 0, 1)$ with $z_{UB} = 47$. Problem RP5 is not pruned.

We assign the index $k = 3$ to these two problems. Since RP5 was not pruned, we proceed with a new iteration of the algorithm.

Third iteration.

We consider problem RP5. By branching on y_3, we create two additional problems: adding the constraint $y_3 = 0$ we get problem RP6, and adding the constraint $y_3 = 1$ we get problem RP7.

Problem RP6

$$\max\ z = 10y_1 + 12y_2 + 15y_3 + 25y_4$$

subject to

$$y_1 = 1$$
$$y_2 = 1$$
$$y_3 = 0$$
$$y_4 = 0 \text{ or } 1$$

Problem RP7

$$\max\ z = 10y_1 + 12y_2 + 15y_3 + 25y_4$$

subject to

$$y_1 = 1$$
$$y_2 = 1$$
$$y_3 = 1$$
$$y_4 = 0 \text{ or } 1$$

We compute the completions and their corresponding objective values for problems RP6 and RP7.

For problem RP6, we have the completion $\mathbf{y}_{UB} = (1, 1, 0, 0)$ with $z_{UB} = 22$. Problem RP6 is pruned by bound, because $z_{UB} < z_{LB} = 40$.

For problem RP7, we have the completion $\mathbf{y}_{UB} = (1, 1, 1, 0)$ with $z_{UB} = 37$. Problem RP7 is pruned by bound, because $z_{UB} < z_{LB} = 40$.

We assign the index $k = 4$ to these two problems.

The 0-1 branch and bound algorithm stops, because there is no problem to be branched out. Therefore, the solution associated with the lower bound $z_{LB} = 40$, that is to say, the candidate solution, is the optimal solution to the 0-1 IP problem: $\mathbf{y}_{UB} = (0, 0, 1, 1)$.

Undoing the change of variables, we obtain the optimal solution to the knapsack problem of the example:

$$x_1^* = 1, \quad x_2^* = 1, \quad x_3^* = 0, \quad x_4^* = 0, \quad z^* = 40.$$

Figure 6.2 summarizes the entire solution sequence in a diagram.

□

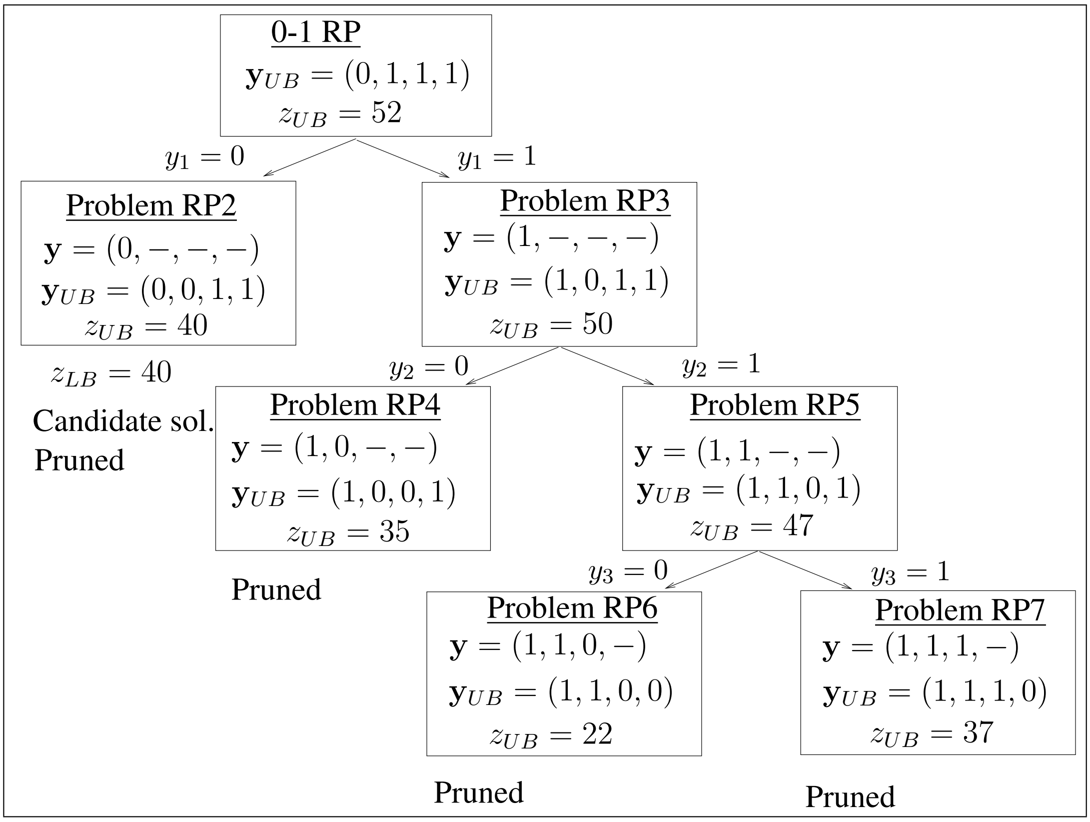


Figure 6.2: Diagram of the entire solution sequence of the knapsack problem of the example.