Integer programming

1 Introduction

One of the most important optmization problems is the linear programming problem (LP), defined in the following way:

$$min \leftarrow c^T x \tag{1}$$

under constraints

$$Ax = b \tag{2}$$

$$x \geq 0 \tag{3}$$

In many practical problems, the variables x should be integer values (e.g. when they describe the number of units in a storage, the number of equipment pieces assigned to a given task, etc.). Then, in the problem defined by (1)-(3) an additional constraint should be taken into account:

$$x_i \in Z \tag{4}$$

where Z denotes the set of integer numbers. Then, the resulting problem, given by (1)-(4) is called an integer programming problem (IP) and (1)-(3) is an associated linear program called the *linear relaxation* (LR).

It can be easily noticed that:

- The optimal objective value for LR is less than or equal to the optimal objective for IP;
- If the solution of the LR problem does not exist, then the solution to the IP problem does not exist, either;
- If the solution of the LP problem satisfies (4), then it is also the solution to the IP problem;
- If all elements of A, B, and c are integers, then the objective value for IP is greater of equal to the objective value for LP rounded up.

Taking into account what has been stated above, it is clear that LR does give some information about the solution of the IP problem; it gives a bound on the optimal value.

2 Branch and Bound method

The solution to the IP problem can be used by iteratively solving appropriate LR problems. At the end of each step, there are three possible cases:

1. The solution to the LR problem consists of integers only. Then it is also the solution to the IP problem
2. At least one of the variables that is a solution to the LR problem is not an integer. Then, the next step should be performed, as described below.
3. The solution to the LR does not exist. Then, the solution to the IP problem does not exist.

In the second case, the IP problem should be decomposed into two subproblems (the *Branch* step). We choose one of the variables, x_j, that is not an integer and, in the first subproblem we add an additional constraint

$$x_j \leq \lfloor k \rfloor, \tag{5}$$

while in the other

$$x_j \geq \lceil k \rceil. \tag{6}$$

where $\lfloor k \rfloor$ denotes the largest integer value not greater than k and $\lceil k \rceil$ denotes the smallest integer value not smaller than k. The *Bound* part of the algorithm will be illustrated by the following example:

$$
\begin{aligned}
&\min \leftarrow f(x) = (8x_1 + 11x_2 + 4x_3) \\
&5x_1 + 7x_2 \geq 5 \\
&3x_1 + 3x_3 \geq 4 \\
&x_3 \geq 1 \\
&x_i \geq 0, \\
&x_i \in Z
\end{aligned}
\tag{7}
$$

The solution to the LR problem is $x_1 = 1$, $x_2 = 0$, $x_3 \approx 0.14$ with $f(x) \approx 8.57$. Hence, the objective for the LR problem is greater or equal to 9. As both x_1 and x_2 are integers, x_3 is the natural choice for the *Branch* step. The resulting subproblems are presented in the Table 1.

Subproblem 1 (P_1)	Subproblem 2(P_2)
$\min (8x_1 + 11x_2 + 4x_3)$	$\min (8x_1 + 11x_2 + 4x_3)$
$5x_1 + 7x_2 \geq 5$	$5x_1 + 7x_2 \geq 5$
$3x_1 + 3x_3 \geq 4$	$3x_1 + 3x_3 \geq 4$
$x_3 \leq 0$ (hence $x_3 = 0$)	$x_3 \geq 1$
$x_i \geq 0$, $x_i \in Z$	$x_i \geq 0$, $x_i \in Z$

Table 1: Decomposition of the problem P into P_1 and P_2

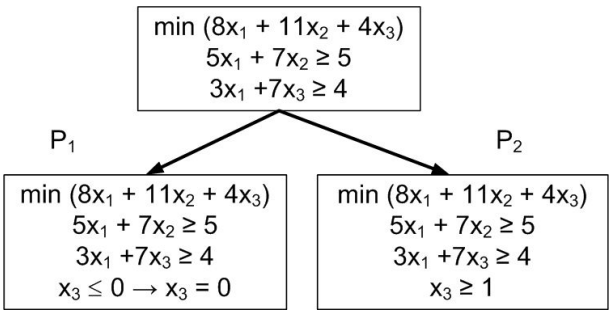

Figure 1: Decomposition of the example (7)

Each of the two subproblems must be dealt with. Let us first tackle the second subproblem (P_2). Its LR solution is $x_1 = 0$, $x_2 \approx 0.71$, $x_3 = 1$ and the objective value $f(x) = 11.86$. Therefore, the objective value of the solution to the IP for this subproblem is at least 12. Variable x_2 is not an integer. Therefore, let us decompose subproblem P_2 with regard to this variable into P_{21} and P_{22} subproblems, shown in the Table 2:

Subproblem (P_{21})	Subproblem (P_{22})
$\min (8x_1 + 11x_2 + 4x_3)$	$\min (8x_1 + 11x_2 + 4x_3)$
$5x_1 + 7x_2 \geq 5$	$5x_1 + 7x_2 \geq 5$
$3x_1 + 3x_3 \geq 4$	$3x_1 + 3x_3 \geq 4$
$x_3 \geq 1$	$x_3 \geq 1$
$x_2 \leq 0$ (hence $x_2 = 0$)	$x_2 \geq 1$
$x_i \geq 0$, $x_i \in Z$	$x_i \geq 0$, $x_i \in Z$

Table 2: Decomposition of the problem P_2 into P_{21} and P_{22}

The solution of the LR for P_21 is $x_1 = 1$, $x_2 = 0$, $x_3 = 1$, with the objective value $f(x) = 12$. Since all variable values are integers, it is also the solution to the IP subproblem P_21. Moreover, the objective value is equal to the lower bound of the solution to the problem P_2. This means, that solving the subproblem P_22 cannot yield better results and we can skip it (unless our goal is finding all minima; for those who want still to check it, it is given by $x_1 = 0$, $x_2 = 1$, $x_3 = 1$ and $f(x) = 15$).

We have now a feasible solution to the IP problem (7). However, there is still an unsolved problem P_1 remaining, for which the lower bound is 9. This means, that following this branch of a tree can lead to a better solution.

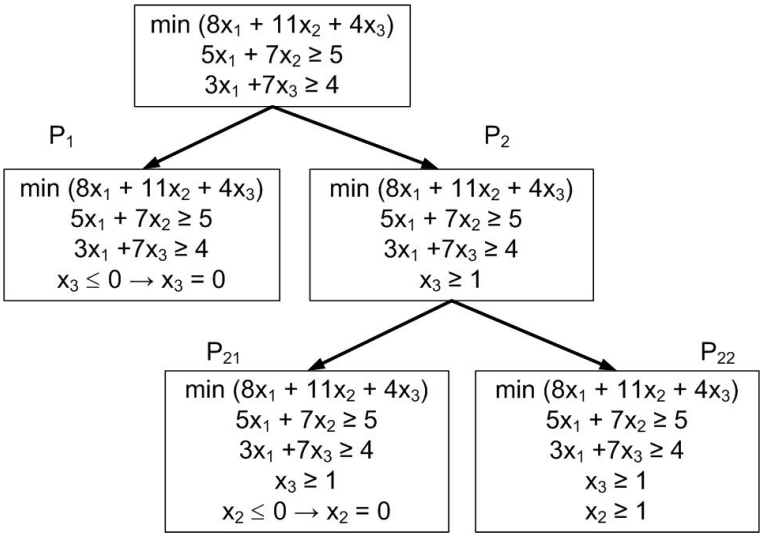

Figure 2: Decomposition of the Subproblem P_2

The solution to the LR for the P_1 is given by $x_1 \approx 1.33$, $x_2 = 0$, $x_3 = 0$, with the objective value $f(x) \approx 10.67$. Therefore, we need to decompose P_1 into two subproblems P_11 and P_12 (see Fig. 3).

Subproblem (P_{11})	Subproblem (P_{12})
$\min (8x_1 + 11x_2 + 4x_3)$	$\min (8x_1 + 11x_2 + 4x_3)$
$5x_1 + 7x_2 \geq 5$	$5x_1 + 7x_2 \geq 5$
$3x_1 + 3x_3 \geq 4$	$3x_1 + 3x_3 \geq 4$
$x_3 = 0$	$x_3 = 0$
$x_1 \leq 1$	$x_1 \geq 2$
$x_i \geq 0$, $x_i \in Z$	$x_i \geq 0$, $x_i \in Z$

The objective value for the solution of LR for P_{12} $f(x) = 16$. Since it is greater (worse) than the best solution obtained so far in other subproblems, this branch will be neglected from now on. In turn, the solution to the LR for P_{11} does not exist. As a result, we can conclude that the solution to the P_{22} problem constitutes the solution to the original IP problem.

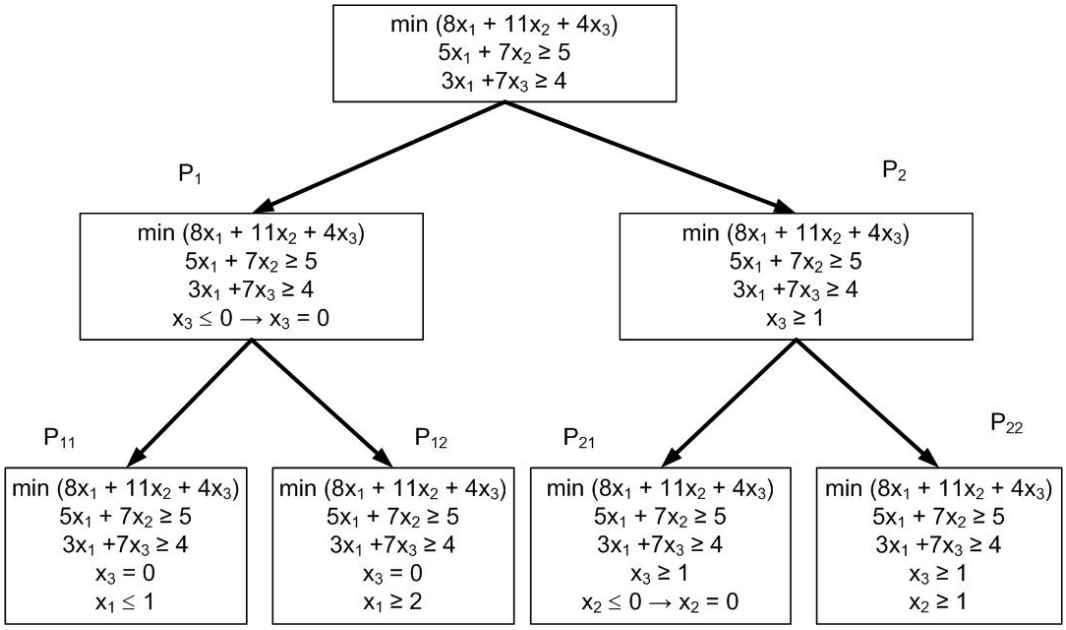

Figure 3: Decomposition of the Subproblem P_1

3 Binary programming

Binary programming is a special case of the integer programming. It can be defined in the following way:

$$\min \leftarrow f(x) = c^T x \tag{8}$$

$$Ax = b \tag{9}$$

$$x_i \in 0, 1 \tag{10}$$

One of the typical examples of the binary programming problems is so called a knapsack problem. We are given N items, each of weight w_j and associated value v_j. We should select items to be put into a knapsack so that their combined value is the largest and at the same time their total weight does not exceed a bound w_{max} imposed on the weight that we can carry.

$$\max \leftarrow v = \sum_{i=1}^{N} x_i v_i \tag{11}$$

$$w = \sum_{i=1}^{N} x_i w_i \leq w_{max} \tag{12}$$

$$x_i \in 0, 1, i = 1, ..., N \tag{13}$$

While the problem can be solved using the Branch and Bound approach presented in the preceding section ("do it yourself" part), the dynamic programming method is worth recalling here.

Let us consider the knapsack problem with $w_{max} = 6$ and the parameters shown in Table 3.

i	v_i	w_i
1	2	3
2	5	2
3	3	4
4	2	1

There are four binary decision variables: x_1, x_2, x_3 and x_4 ($x_i = 1$ and $x_i = 0$ correspond to decisions "to pack" and "not to pack" the item i into the knapsack, respectively). Application of the dynamic programming method leads to four stages in which only one decision variable is considered. The initial state corresponds to an empty knapsack

($v = 0$, $w = 0$). Let us denote by S_{ij} the state of the system after making the decision at stage i: S_{i1} and S_{i0} represent the states resulting from $x_i = 1$ and $x_i = 0$, respectively. Similarly, the change in the combined weight w_{ij} and value v_{ij} are defined..

The first step of the dynamic programming is shown in the Fig. 4. If the first item is put into the knapsack, $w_{11} = w + w_1 = 3$ and $v_{11} = v_1 + v = 2$. Otherwise $w_{10} = w = 0$ and $V_{10} = v = 0$.

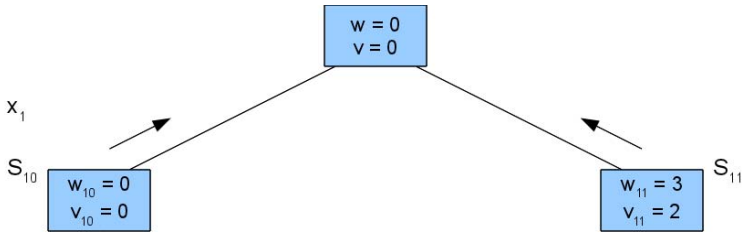

Figure 4: Solving the knapsack problem - the first step

At the second stage, the state depends on what has been done previously. If the second item is put into the knapsack, the combined value and weight will increase by 5 and 2, respectively. However, the state S_{21} can be reached either from S_{11} or S_{10}, yielding $w*_{21} = w11 + 2 = 5$, $v*_{21} = v_{11} + 5 = 7$ or $w*_{21} = w_{10} + 2 = 2$, $v*_{21} = v_{10} + 5 = 5$ in the former and latter case, respectively. Since our goal is to maximize the combined value of items in the knapsack, we should go from state S_{21} to S_{11} and not to S_{10}. Therefore, $w_{21} = 5$, $v_{21} = 7$. Similarly, having analyzed S_{20}, we obtain $w_{20} = 3$, $v_{20} = 2$ (see Fig. 5).

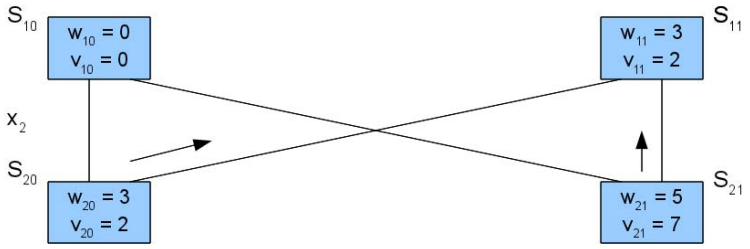

Figure 5: Solving the knapsack problem - the second step

The third and the fourth step of the decision process are illustrated in Figs 6 and 7, correspondingly.

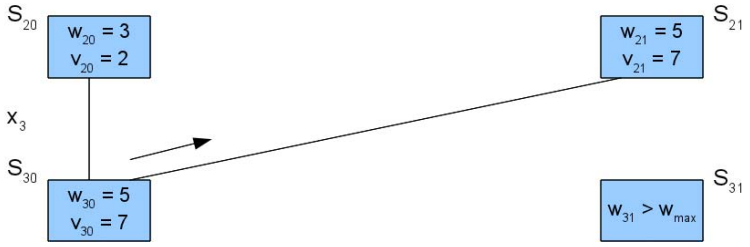

Figure 6: Solving the knapsack problem - the third step

4 Laboratory exercise

The goal of this exercise is to get familiar with algorithms developed for solving IP problems (including binary IP). At the beginning, students take a simple written or oral test, yielding maximum of 1 point. Afterwards, they are divided into subsections (one or two persons in each subsection) and solve the assigned knapsack and IP problems for a maximum of 4 points. The knapsack problem should be solved twice: using branch and bound an dynamic programming methods (each solution is worth 1 point). At the end of the classes students present their solutions (complete description of each step is required) and are graded. These solutions are regarded as the reports. To get credit for the exercise, one should get at least 3 points (combined).

To solve the LR of the IP problem, students should use software that is available.

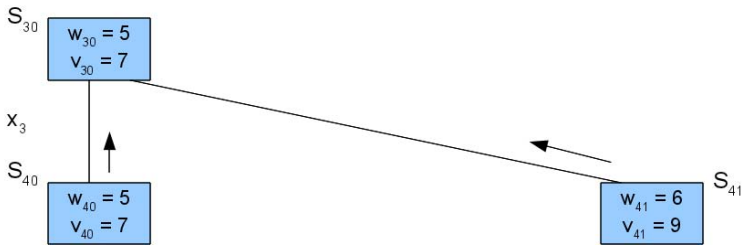

Figure 7: Solving the knapsack problem - the fourth step

5 Exemplary tasks

Task 1. Find

$$\min \leftarrow 3x_1 + 2x_2 + x_3 \tag{14}$$

under constraints

$$x_1 + 2x_2 \geq 4 \tag{15}$$
$$3x_1 + x_3 \geq 8 \tag{16}$$
$$x_i \geq 0, x_i \in Z, i = 1, 2, 3. \tag{17}$$

Task 2.
Find

$$\min \leftarrow x_1 + x_2 + 2x_3 \tag{18}$$

under constraints

$$2x_1 + 3x_2 \geq 5 \tag{19}$$
$$x_2 + 2x_3 \geq 4 \tag{20}$$
$$x_i \geq 0, x_i \in Z, i = 1, 2, 3. \tag{21}$$

Task 3.
Find the solution to the knapsack problem for $w_{max} = 6$ and the following parameters:

i	v_i	w_i
1	5	4
2	3	3
3	4	2
4	3	2

Task 4.
Find the solution to the knapsack problem for $w_{max} = 4$ and the following parameters:

i	v_i	w_i
1	4	3
2	2	1
3	3	2
4	3	3