Mixed-Integer Programming

Alberto Santini

Fall 2024

Many examples in these lecture notes are adapted from popular books:

- Alexander Schrijver (1998). *Theory of linear and integer programming*. Wiley. ISBN: 0-471-98232-6.
- Vašek Chvátal (1983). *Linear Programming*. W.H. Freeman and Company. ISBN: 0-716-71195-8.
- Laurence Wolsey (2020). *Integer Programming*. 2nd Edition. Wiley. ISBN: 978-1-119-60653-6.
- Silvano Martello and Paolo Toth (1990). *Knapsack Problems: algorithms and computer implementations*. Wiley. ISBN: 978-0-471-92420-3.

Recall that a Linear Programme (LP) in standard form is a mathematical model of the following type:

$$\max \quad \sum_{j=1}^{n} c_j x_j \tag{1a}$$

$$\text{subject to} \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad \forall i \in \{1, \dots, m\} \tag{1b}$$

$$x_j \geq 0 \qquad \forall j \in \{1, \dots, n\}. \tag{1c}$$

All variables in an LP are continuous, even when considering the non-standard form with non-positive or free variables. While many interesting problems can be modelled as LPs, many more require non-continuous (i.e., discrete-valued) variables x_j. For example, if x_j counts the number of trains to schedule between two cities on a given day, it should take integer values. If x_j denotes the decision of opening or not a factory, it should take a false/true (or, in mathematical terms, 0/1) value.

To model problems requiring non-continuous variables, we introduce Mixed-Integer Linear Programmes (MILPs). These models feature both continuous and discrete variables. Like LPs, they require that the objective function and the constraints are linear. The standard form of a MILP reads as follows.

$$\max \quad \sum_{j=1}^{n} c_j x_j + \sum_{j=1}^{q} d_j y_j \tag{2a}$$

$$\text{subject to} \quad \sum_{j=1}^{n} a_{ij} x_j + \sum_{j=1}^{q} e_{ij} y_j \leq b_i \qquad \forall i \in \{1, \dots, m\} \tag{2b}$$

$$x_j \geq 0 \qquad \forall j \in \{1, \dots, n\} \tag{2c}$$

$$y_j \geq 0 \text{ and integer} \qquad \forall j \in \{1, \dots, q\}. \tag{2d}$$

The x variables are continuous and non-negative, as required by domain definition (2c); they are called the **continuous variables**. The y variables are integer and non-negative, as required by domain definition (2d); they are called the **integer variables**. Both continuous and integer variables can appear in the objective function (2a) and the constraints (2b).

As for LPs, we do not limit ourselves to the standard form when writing down a MILP. In the following, we will take the freedom of using $\geq$-inequalities and equalities and let the y variables take values in arbitrary discrete intervals. For example, we can write that $y_j \in \{4, \dots, 8\}$. This variable domain definition is equivalent to the standard "$y_j \geq 0$ and integer" and the constraints $y_j \geq 4$ and $y_j \leq 8$. In particular, when we impose that $y_j \in \{0, 1\}$, we call y_j a **binary variable**.

The nomenclature is not fully standardised in the scientific literature, but, as a common rule, we talk about:

- Mixed-Inger Linear Programmes when both continuous and integer variables are present.
- Integer Linear Programmes when all variables are integer.
- Binary Linear Programmes when all variables are binary.

We often omit the word "Linear" when it is clear from the context, such as in these lecture notes. According to the above definitions, all binary programmes are integer, and all integer programmes are mixed-integer; none of the reverse implications is true.

The integer variables of a MILP will be of particular interest. Therefore, in the remainder of this section, we will often consider Integer Programmes (IPs) rather than MILPs. Let the standard form of an IP be the following:

$$\max \quad \sum_{j=1}^{n} c_j x_j \tag{3a}$$

$$\text{subject to} \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad \forall i \in \{1, \dots, m\} \tag{3b}$$

$$x_j \geq 0 \text{ and integer} \qquad \forall j \in \{1, \dots, n\}. \tag{3c}$$

1 The feasible region of an IP

Consider the following IP with two variables:

$$\max \quad x_1 + x_2 \tag{4a}$$

$$\text{s.t.} \quad x_1 \leq 6 \tag{4b}$$

$$50x_1 + 81x_2 \leq 561 \tag{4c}$$

$$10x_1 + 3x_2 \leq 55 \tag{4d}$$

$$x_2 \leq 5 \tag{4e}$$

$$x_1, x_2 \geq 0 \text{ and integer.} \tag{4f}$$

Let us disregard for a moment the integrality constraint on the variables and replace (4f) with $x_1, x_2 \geq 0$. The corresponding formulation would be an LP. Indeed, any time we transform a MILP by omitting the integrality constraint, we obtain it **continuous relaxation**. The continuous relaxation is always an LP; it shares the same objective function and constraints as the original MIP, but its variables are continuous, i.e., their domain definition only includes the non-negativity constraints without the integrality requirement. Figure 1 shows the feasible

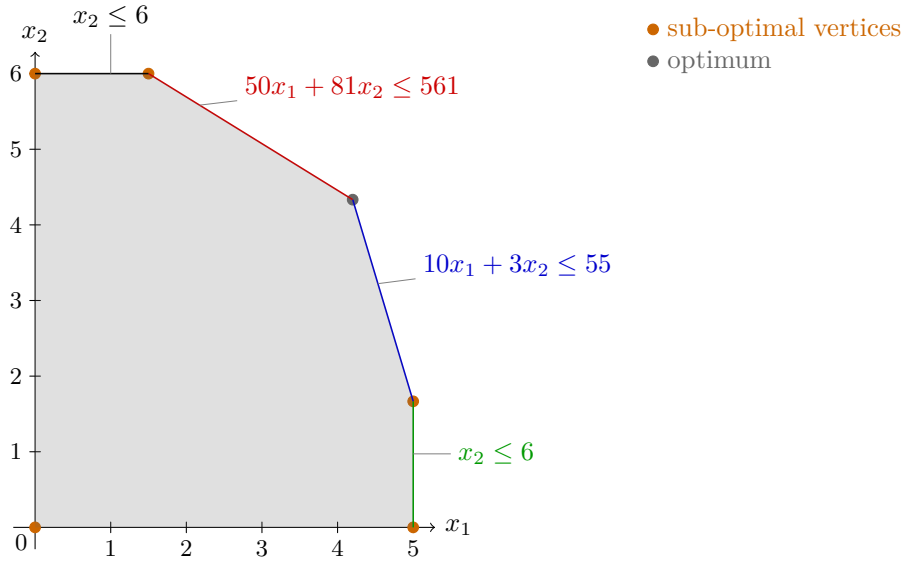

Figure 1: Feasible region of the continuous relaxation of (4a)–(4f).

region of the continuous relaxation of (4a)–(4f). The polytope has six vertices and a unique optimum at $(\frac{21}{5}, \frac{13}{3})$, with objective value $8.5\bar{3}$.

We remark that Figure 1 does *not* show the feasible region of the original IP (4a)–(4f). Indeed, in the IP, variables x_1 and x_2 can only take integer values and any point with fractional coordinates is not part of the feasible region. Figure 2 shows the feasible region of the IP. This problem admits two optima at $(3, 5)$ and $(4, 4)$, with objective value 8.

Let us denote with P the feasible region of an IP such as (3a)–(3c) and with P' the feasible region of its continuous relaxation. Region P is a grid of points with integer coordinates that satisfy constraints (3b). Therefore, P is a disconnected non-convex set; by contrast, P' (as all feasible regions of LPs) is connected and convex.

Moreover, P is a subset of P'; in particular, it is the intersection of P' and the integer lattice $\mathbb{N}^n$ (recall that n is the number of variables). It follows that the optimal solution of an IP in maximisation form cannot be larger than the optimal solution of its continuous relaxation simply because the IP's feasible region is smaller. Any point feasible for the IP is feasible for the continuous relaxation. However, not all points feasible for the continuous relaxation are feasible for the IP. Among them, some might have a better objective value than the IP optimum.

For example, as we have seen before, the optimal solution of the continuous relaxation of (4a)–(4f) has a value of $8.5\bar{3}$. Still, the optimum of the IP has a value of 8.

Still, we might be tempted to use the following approach to solve an IP. First, we consider its continuous relaxation. Because this is an LP, we know how to solve it. If all solution components are integer, we have also solved the original IP (why?). Otherwise, we could round each component up or down to its nearest integer value and, thus, obtain an integer solution. What is the problem with this approach? There are at least two.

1. No rounding of the LP optimum might be feasible for the IP. Figure 3 shows such an example: the continuous relaxation optimum if $(\frac{9}{4}, \frac{3}{2})$, while the IP optimum is $(1, 1)$. The four possible roundings of the continuos relaxation optimum are $(2, 1)$, $(2, 2)$, $(3, 2)$ and $(3, 1)$. None of them is feasible for the IP.

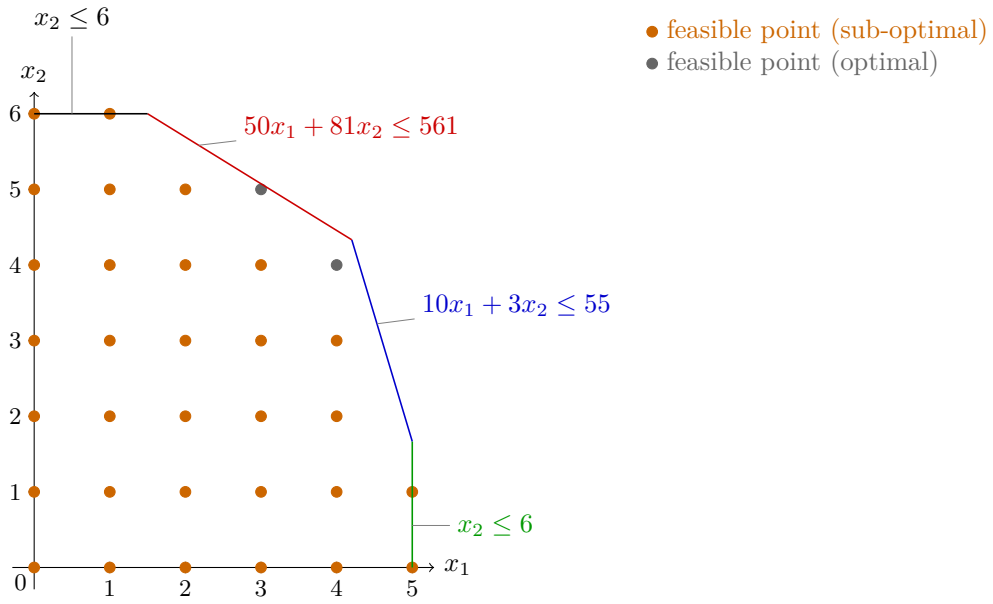

Figure 2: Feasible region of (4a)–(4f).

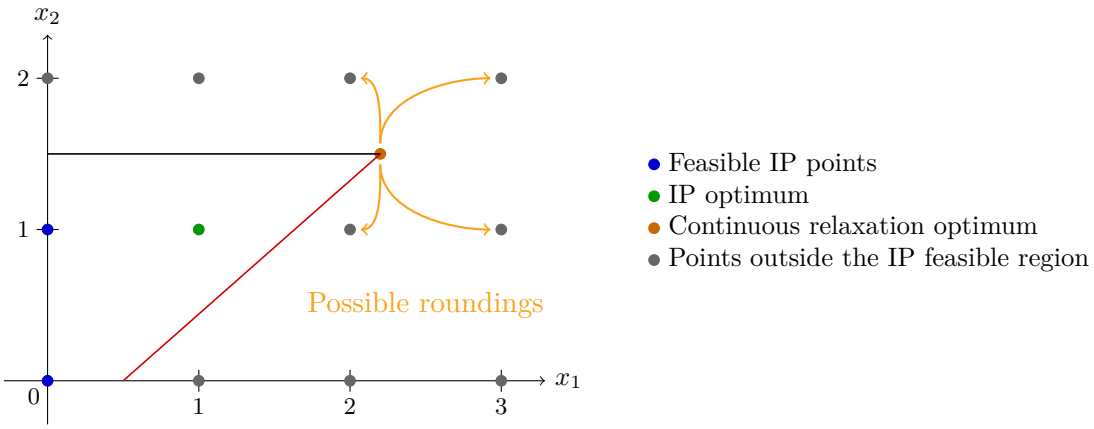

Figure 3: Example of an IP whose continuous relaxation's optimum admits no IP-feasible rounding.

2. There is no guarantee that the continuous relaxation optimum lies near the IP optimum, nor that its objective value is close to the IP optimum's. Indeed, in Exercise 1, you will show that it is possible to construct an IP such that its continuous relaxation is arbitrarily bad.

Exercise 1. Write down a family of simple maximisation IPs parameterised by some $h \in \mathbb{R}^+$, with the following property. As h grows, the difference between the objective value of the optimal solution of the continuous relaxation and the objective value of the optimal IP solution grows larger. Eventually, the difference tends to infinity when $h \to \infty$.

2 Solving IPs

As we have seen, deriving the solution of an IP from its continuous relaxation is not straightforward. Still, the continuous relaxation is basically the only problem closely related to the original IP that we know how to solve (because it is an LP). Therefore, looking for algorithms that exploit the continuous relaxation to solve the IP seems natural.

Historically, the first algorithm to use this idea was the **Cutting Plane** method. This iterative method starts with solving the continuous relaxation of the IP. While the optimal solution is not integer, it adds new constraints to the formulation. Each new constraint is such that (i) no feasible solution of the IP violates it, but (ii) some fractional solutions, including the optimum of the continuous relaxation, violate it. Because of the second property, the optimum will be different when solving the continuous relaxation again with the new constraint. At the same time, due to the first property, no integer feasible solution is ever removed. It is possible to prove that, carefully selecting the new constraints to add, this algorithm terminates, i.e., the optimal solution of the continuous relaxation eventually becomes a feasible IP point in a finite number of steps. Despite this property, the cutting plane algorithm's practical adoption is extremely limited nowadays due to its poor computational performance.

The *de facto* standard algorithm to solve IPs is, instead, the **branch-and-bound** (BB) algorithm. We describe this algorithm in the following section.

3 The branch-and-bound algorithm

We introduce the BB algorithm with an example considering the following IP, also depicted in Figure 4.

$$\begin{aligned} \max \quad & 2x_1 + 3x_2 && \text{(5a)} \\ \text{subject to} \quad & 2x_2 \geq 1 && \text{(5b)} \\ & x_1 - 7x_2 \geq -21 && \text{(5c)} \\ & 3x_1 + x_2 \leq 14 && \text{(5d)} \\ & x_1, x_2 \geq 0 \text{ and integer.} && \text{(5e)} \end{aligned}$$

The unique optimum of the continuous relaxation is point $x^* = (\frac{7}{2}, \frac{7}{2})$, with objective value $z^* = \frac{35}{2} = 17.5$. The BB algorithm starts from the following observation: we do not know the IP optimum, but its coordinates (i.e., the values taken by its variables) must be integer by definition. Therefore, our current point x^* is undoubtedly not the IP optimum because it has at least one fractional coordinate (in our case, both). For example, it is impossible that $x_1 = \frac{7}{2} = 3.5$ in any integer feasible solution. On the other hand, if we solved the continuous

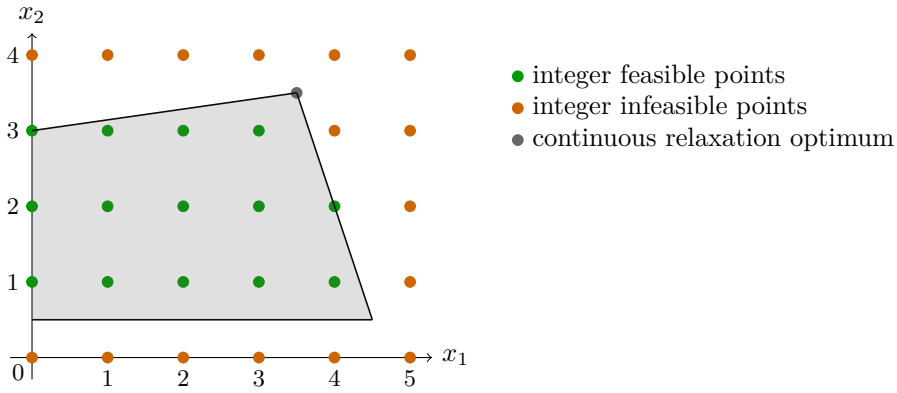

Figure 4: Feasible region of an IP (green points) and its continuous relaxation (shaded area). The grey point is the continuous relaxation optimum.

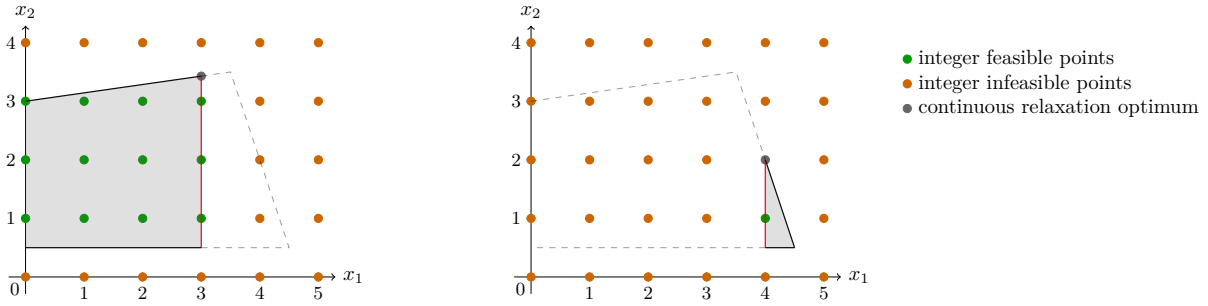

Figure 5: Feasible regions of the sub-problems obtained imposing $x_1 \leq 3$ (left) or $x_1 \geq 4$ (right).

relaxation and the corresponding optimum had all integer coordinates, it would also be the IP optimum.

Because integer feasible points must have an integer value for variable x_1, the following disjunction is trivially true:

$$x_1 \leq 3 \quad \text{or} \quad x_1 \geq 4. \tag{6}$$

If we added this disjunction to our formulation, we would immediately "cut out" the current LP optimum x^* because it violates both inequalities. However, (6) is not a valid IP constraint. Indeed, it is a disjunction of two linear inequalities. Linear inequalities are allowed in an IP (or an LP), but disjunctions of inequalities are not.

We solve this issue by creating two copies of the original formulation (5a)–(5e). We add inequality $x_1 \leq 3$ in the first copy and inequality $x_1 \geq 4$ in the second copy, as shown in Figure 5. The feasible region of the two resulting formulations is smaller than the original. We are sure the integer optimum must be in one of the IP feasible regions of the new *subproblems* because the union of these two IP feasible regions gives the original IP feasible region. Indeed, we leave out points with $3 < x_1 < 4$, i.e., with a strictly fractional coordinate, and therefore we are not removing any integer feasible point. Yet, we don't know which of the two copies (i.e., which of the two feasible regions) will contain the integer optimum; we must explore both feasible regions.

The first main idea behind the BB algorithm is to recursively repeat the solution method we used for formulation (5a)–(5e). First, let us focus on the subproblem associated with constraint $x_1 \leq 3$. Its complete formulation is

$$\max \quad 2x_1 + 3x_2 \tag{7a}$$

$$\begin{aligned} \text{subject to} \quad & 2x_2 \geq 1 && \text{(7b)} \\ & x_1 - 7x_2 \geq -21 && \text{(7c)} \\ & 3x_1 + x_2 \leq 14 && \text{(7d)} \\ & x_1 \leq 3 && \text{(7e)} \\ & x_1, x_2 \geq 0 \text{ and integer.} && \text{(7f)} \end{aligned}$$

If we solve the continuous relaxation of this subproblem, we obtain the optimum $x^* = (3, \frac{24}{7})$ with objective value $z^* = \frac{120}{7} \approx 17.14$. Because variable x_2 takes the fractional value $\frac{24}{7} \approx 3.43$, x^* is not integer feasible. Indeed, any integer feasible point will satisfy the following disjunction:

$$x_2 \leq 3 \quad \text{or } x_2 \geq 4.$$

Therefore, we can further divide the feasible region depicted in the left part of Figure 5 and create two more copies of the original formulation. The first copy will contain the additional constraints $x_1 \leq 3$ and the new $x_2 \leq 3$, while the second copy will contain $x_1 \leq 3$ and the new $x_2 \geq 4$.

Let us, for a moment, leave these two new "sub-copies" aside and examine the other open subproblem, i.e., the one associated with constraint $x_1 \geq 4$. The complete formulation of this problem is

$$\begin{aligned} \max \quad & 2x_1 + 3x_2 && \text{(8a)} \\ \text{subject to} \quad & 2x_2 \geq 1 && \text{(8b)} \\ & x_1 - 7x_2 \geq -21 && \text{(8c)} \\ & 3x_1 + x_2 \leq 14 && \text{(8d)} \\ & x_1 \geq 4 && \text{(8e)} \\ & x_1, x_2 \geq 0 \text{ and integer.} && \text{(8f)} \end{aligned}$$

The feasible region of this IP and of its continuous relaxation is depicted in the right part of Figure 5. The optimum of the continuous relaxation is $x^* = (4, 2)$, which is an integer feasible point. Congratulations: we have found the optimal solution of (8a)–(8f). Point $(4, 2)$ is integer feasible and has objective value $z^* = 14$. Still, this point is optimal only in the restricted subproblem with the added constraint $x_1 \geq 4$. It is still an open problem whether this point will also be optimal for the original (5a)–(5e) or if, otherwise, there is an even better point contained in the other feasible subregion, the one associated with (7a)–(7f).

To answer this question, we have to explore the two open subproblems:

$$\begin{aligned} \max \quad & 2x_1 + 3x_2 && \text{(9a)} \\ \text{subject to} \quad & 2x_2 \geq 1 && \text{(9b)} \\ & x_1 - 7x_2 \geq -21 && \text{(9c)} \\ & 3x_1 + x_2 \leq 14 && \text{(9d)} \\ & x_1 \leq 3 && \text{(9e)} \\ & x_2 \leq 3 && \text{(9f)} \\ & x_1, x_2 \geq 0 \text{ and integer.} && \text{(9g)} \end{aligned}$$

$$\begin{aligned} \max \quad & 2x_1 + 3x_2 && \text{(10a)} \\ \text{subject to} \quad & 2x_2 \geq 1 && \text{(10b)} \\ & x_1 - 7x_2 \geq -21 && \text{(10c)} \\ & 3x_1 + x_2 \leq 14 && \text{(10d)} \\ & x_1 \leq 3 && \text{(10e)} \\ & x_2 \geq 4 && \text{(10f)} \\ & x_1, x_2 \geq 0 \text{ and integer.} && \text{(10g)} \end{aligned}$$

These subproblems are depicted in Figure 6. Let us first deal with the straightforward case of (10a)–(10g): the new constraint $x_2 \geq 4$ defines a hyperplane completely disjoint from the original feasible region. Therefore, the feasible region of (10a)–(10g) is empty, and the subproblem is

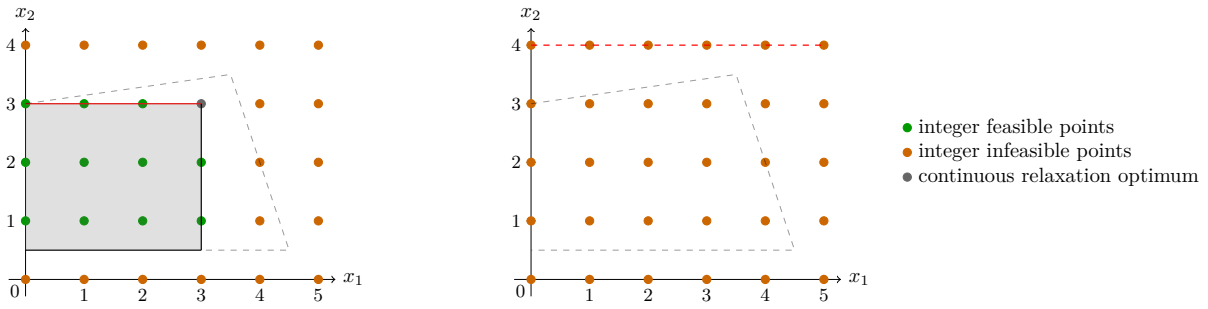

Figure 6: Feasible regions of the sub-problems obtained imposing $x_1 \leq 3$ and $x_2 \leq 3$ (left) or $x_1 \leq 3$ and $x_2 \geq 4$ (right).

infeasible. Conversely, the feasible region of (9a)–(9g) is non-empty. Solving its continuous relaxation, the unique optimum is $x^* = (3, 3)$. Because this point has all integer coordinates, it is also optimum for the integer problem and, thus, (9a)–(9g) is solved to optimality. The objective value of this optimum is $z^* = 15$, which is higher than the objective value of the other integer feasible point that we have previously examined (this was point $(4, 2)$ with objective value 14).

Let us summarise what we have done until now. First, we have divided the original feasible region into two parts, one containing solutions such that $x_1 \leq 3$ and the other containing solutions such that $x_1 \geq 4$. We found the optimal integer solution for the second part, which had an objective value of 14. Then, we continued exploring the first part and further divided it into two subparts, one containing solutions such that $x_2 \leq 3$ and the other containing solutions such that $x_2 \geq 4$. We realised that the integer feasible region of the second subpart was empty, so we had to explore the first subpart. We found the optimal integer feasible solution for that subpart, with objective value 15.

Schematically, we explored the feasible region as shown in Figure 7. The original integer feasible region is the union of all the smaller regions we have explored. Therefore, by taking the best integer optimum over all the smaller regions, we obtain the overall optimum of (5a)–(5e).

3.1 The branch-and-bound tree

An alternative schematic depiction of the way we explored the feasible region of (5a)–(5e) is provided in Figure 8. The structure in this figure is known as a **branch-and-bound tree** and each box (which, in our example, contains a reference to a formulation) is called a **node**. The tree's root node represents the original problem with its entire feasible region. The subproblems of the original (5a)–(5e) obtained by adding inequalities $x_1 \leq 3$ and $x_1 \geq 4$ are represented as children of the root node. We draw arrows linking the parent to the child nodes and annotate them, specifying which inequality we add. Further subdivisions give rise to new nodes at deeper levels of the tree. We also annotate each node by recording the optimum x^* of its continuous relaxation, the corresponding objective value z^* attained, and whether the optimum is integer. Remark that when the optimum is integer, we need not add any new inequality and the corresponding node has no children. Such a node is called a **leaf node**.

3.2 Branching

We add new inequalities when at least one coordinate of the optimum of the continuous relaxation at a node is fractional. The procedure that creates two child nodes from such a parent node

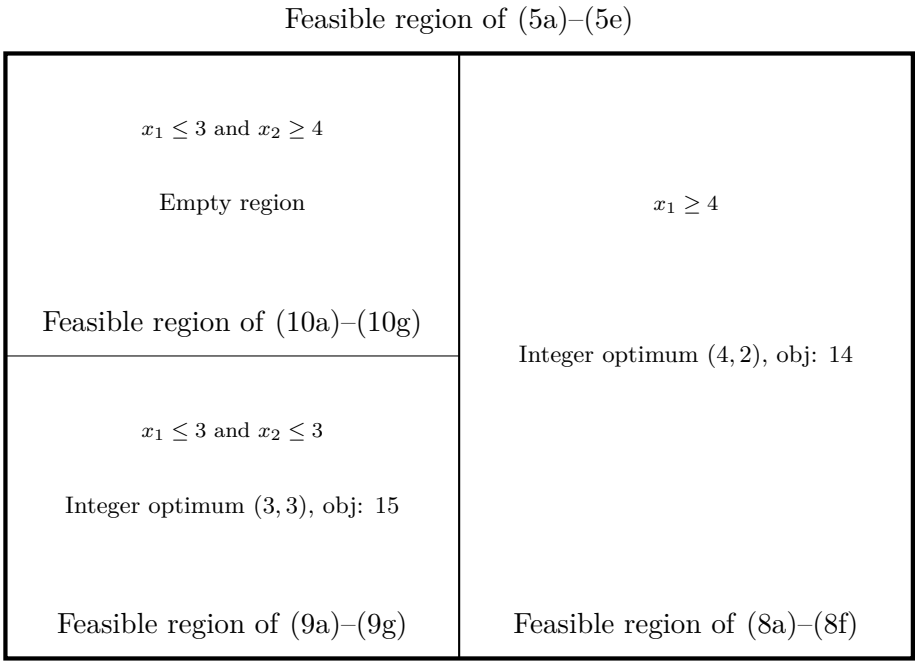

Figure 7: Schematic representation of the feasible region, as we explored it.

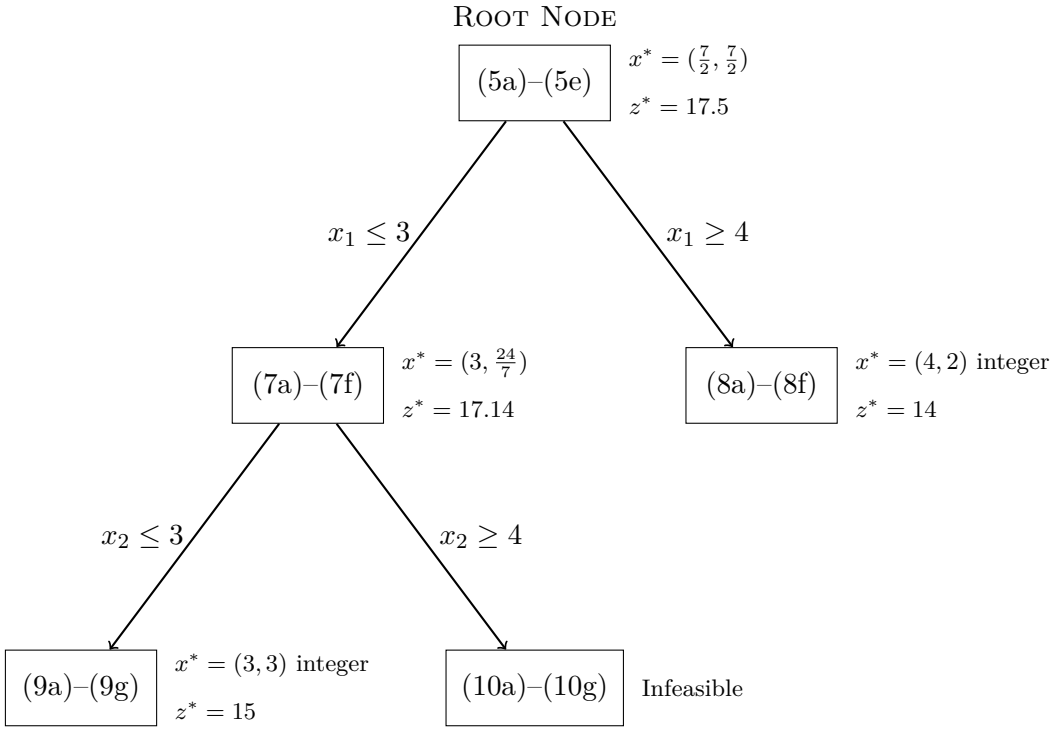

Figure 8: Tree representation of our exploration of the feasible region of (5a)–(5e).

is called **branching**. If x^* has more than one fractional component, we must choose the one we use for branching. For example, at the root node $x^* = (\frac{7}{2}, \frac{7}{2})$, i.e., both x_1 and x_2 were fractional, and we arbitrarily decided to add inequalities involving variable x_1. The chosen fractional variable is known as the **branching variable**.

If x_j is the branching variable and x_j^* is the value it takes in x^*, we add one of the following constraints to each of the two children nodes:

$$x_j \leq \lfloor x_j^* \rfloor \quad \text{and} \quad x_j \geq \lceil x_j^* \rceil.$$

When there are multiple potential branching variables, instead of operating an arbitrary choice as we have done, the BB algorithm follows a **branching strategy**. This strategy comprises a set of rules that determine which fractional variable will become the branching one. Historically, BB implementations used rules of thumb such as selecting the most fractional variable (the one whose fractional part is closest to 0.5) or, conversely, choosing the least fractional one (the one whose fractional part is close to 0 or 1). Modern BB implementations use more sophisticated rules. Some involve training a machine learning algorithm on the first nodes explored and then using it to predict good branching variables when exploring the rest of the nodes.

3.3 Bounding

If the feasible region of the IP is bounded, the algorithm we have described until now is guaranteed to end in a finite number of steps. This formal result, which we do not present, has an intuitive explanation. At each step, we either prove that a node is infeasible or divide its feasible region into smaller ones until, eventually, the optimum of the continuous relaxation is integer. Still, if we followed this algorithm, the number of nodes in the tree might grow extremely large. The real strength of the BB algorithm comes from the combination of the branching and the **bounding** phases. Bounding is used to prove that entire parts of the tree cannot possibly contain the optimum and can thus be pruned.

To understand how bounding works, we must first make a few observations.

- Every time we obtain an integer solution x^*, its objective value z^* is a lower bound on the unknown objective value z^{opt} of the optimal solution of the IP. Remark that x^* is a feasible solution, and our IP is in maximisation form. By definition, the optimum cannot be worse than another feasible solution and, therefore, $z^* \leq z^{\text{opt}}$. If our IP were in minimisation form, z^* would be an upper bound on z^{opt} (i.e., $z^{\text{opt}} \leq z^*$). To avoid confusion, it is useful to call the bound from an integer feasible solution the **primal bound**. The primal bound is a lower bound for a maximisation problem and an upper bound for a minimisation problem.
- Consider the continuous relaxation of the IP at the root node and assume that its optimal solution—or *all* its optimal solutions, if there are multiple—are fractional. Let x^* be such a solution; its objective value z^* is an upper bound on the unknown value z^{opt} of the optimal solution of the IP. Indeed, the continuous relaxation uses the same objective function and constraints of the IP but relaxes the integrality requirements. Therefore, its feasible region contains the feasible region of the IP. It follows that the IP optimum must be contained inside the continuous relaxation of the feasible region. Therefore, the optimal solution of the continuous relaxation cannot be worse than the optimal solution of the IP, i.e., $z^{\text{opt}} \leq z^*$. Analogously to what we discussed in the previous point, the value of z^* would instead be a lower bound on z^{opt} if we dealt with a minimisation problem. To avoid confusion, we call the bound coming from the optimal solution of the continuous relaxation

of the problem the **dual bound**. The dual bound is an upper bound for a maximisation problem and a lower bound for a minimisation problem.

- The role of the dual bound can be generalised to nodes other than the root node. Suppose the optimal solution of the continuous relaxation of a subproblem is fractional. In that case, its objective value is a dual bound on the objective value of that subproblem's best integer feasible solution. In other words, there is no integer feasible solution with a strictly better objective value in the subtree rooted at the node.

The above considerations suggest a strategy to prune (hopefully) large portions of the BB tree. Let $\underline{z}$ be the objective value of the best integer feasible solution encountered during the tree exploration up to the current node. Because of our first observation, $\underline{z}$ is a primal bound on z^{opt}. For example, $\underline{z}$ is a lower bound for a maximisation problem. Because $\underline{z}$ is the objective value of *the best* feasible solution encountered, it is the tightest lower bound available. If a better integer feasible solution is encountered later, the value of $\underline{z}$ should be updated accordingly. It is also convenient to keep track of the corresponding integer feasible solution $\underline{x}$ that produced the bound $\underline{z}$.

Now, assume that we are exploring a given BB node. We solve the corresponding subproblem's continuous relaxation and find a possibly fractional optimal solution x^* with objective value z^*. If z^* is worse than $\underline{z}$ (i.e., if $z^* < \underline{z}$ for a maximisation problem), then **the subtree rooted at the current node cannot possibly contain the IP optimum and should be pruned**. Indeed, the best possible objective value of a solution in the subtree is z^*. But still, this value is worse than the objective value of the best known integer feasible solution, $\underline{z}$. Therefore, all integer feasible solutions that we can possibly visit if we keep exploring that subtree are "uninteresting" because they cannot be as good as $\underline{x}$. This entire subtree is then superfluous for our task and can be ignored entirely. The following exercise gives a practical insight into the role of bounding.

Exercise 2. Solve the following IP via branch-and-bound. If the optimal solution of the continuous relaxation at some node has more than one fractional variable, branch on the variable with the smallest index. Because this IP only has two variables, this rule means that you should branch on x_1 if both x_1 and x_2 are fractional. If multiple open nodes exist, explore the one higher up in the tree. In case of ties, explore the node created last; if there still is a tie, explore the $\leq$ node first.

$$\begin{aligned}
\max \quad & 3x_1 + 5x_2 && \text{(11a)}\\
\text{subject to} \quad & 2x_1 + 4x_2 \leq 13 && \text{(11b)}\\
& x_1 \leq 4 && \text{(11c)}\\
& x_2 \leq 3 && \text{(11d)}\\
& x_1, x_2 \geq 0 \text{ and integer.} && \text{(11e)}
\end{aligned}$$

3.4 Tree exploration

In Section 3.2, we have discussed a critical design decision when implementing a BB algorithm: how to choose the branching variable. Here, we discuss another: choosing the next open node to explore when there are multiple, i.e., which **tree exploration strategy** to employ. Traditionally, there are two popular strategies, which we describe in the following.

3.4.1 Depth-first exploration

With the **depth-first** strategy, we prefer exploring nodes further down in the tree, i.e., resulting from more nested branching constraints. If the problem is feasible, it is easier to find an integer feasible solution when using this strategy. The idea is that the more branching constraints we impose, the more likely it is that the optimal solution of a continuous relaxation will be integer, thus producing a feasible solution for the original IP. Therefore, this strategy is often used when finding a feasible solution to an IP is challenging. Such problems arise when the IP contains many complicated constraints that are non-trivially satisfiable simultaneously.

A classic example is an exam timetabling problem at a university, in which a decision maker must assign each course a time slot and a classroom for the final exam. Because all exams are concentrated in the same week, the number of classrooms is limited, and students cannot take too many exams in a short time, finding even one feasible solution to this problem can be very hard.

3.4.2 Best-first exploration

With the **best-first** strategy, we prefer exploring the node whose father has the loosest dual bound. For example, for a maximisation problem, we choose the nodes whose father provides the largest upper bound. Because the dual bounds get tighter and tighter as we go down the tree (because we add more and more constraints), these nodes tend to be near the tree top.

Let us call the "father bound" the dual bound of the father of an open node. Remark that if we were time-constrained and had to stop the BB algorithm while there are open nodes, the largest father bound would provide a dual bound on the overall original IP. The reason is that each father bound limits what is the best objective value achievable in a given subtree. A priori, we do not know which subtree will contain the IP optimum. Therefore, we must pessimistically choose the loosest father bound as the overall dual bound on z^{opt}. By exploring the node with the loosest father bound, we can tighten this overall bound quickly.

This strategy is especially useful if we believe we have a very good (perhaps optimal) integer feasible solution, and we must then focus on *proving* its optimality. In the extreme case in which our current best integer solution is optimal, the depth-first strategy is ineffective: by definition, we cannot find any strictly better integer feasible solution. Therefore, we should try to close as many open nodes as possible to tighten the IP's overall dual bound; because the node with the highest father bound yields the overall dual bound, that node should be our first target.

This strategy is best suited for problems for which building feasible solutions is easy, and there are known methods for finding high-quality integer feasible solutions. A typical example is the Travelling Salesman Problem, which we will study in more depth later in this course. This problem asks in which order a salesman should visit n cities and return to his origin, minimising the travel costs. For example, visiting Barcelona, Zaragoza, Madrid, Sevilla, Murcia, Valencia and going back to Barcelona makes more sense than hopping all over the place with a tour such as Barcelona → Sevilla → Zaragoza → Murcia → Madrid → Barcelona. Because there are many well-known methods to build almost optimal tours for the Travelling Salesman Problem, one could initialise the "Current best integer solution" with one such method instead of waiting to encounter the first integer solution during the BB tree exploration.

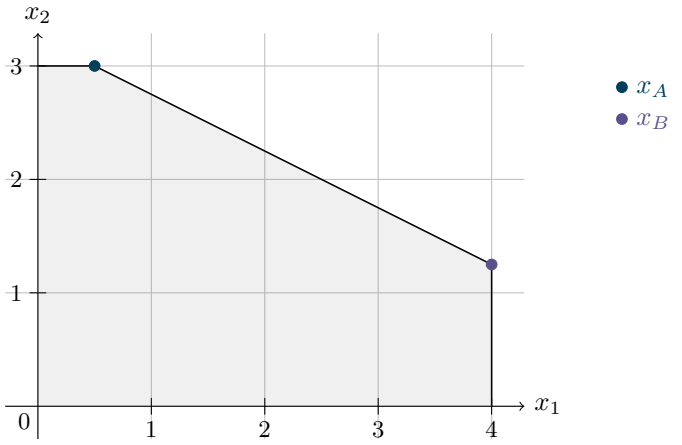

Figure 9: Feasible region and relevant vertices of the continuous relaxation of the IP (11a)–(11e).

4 Solutions

Exercise 1

Consider the following family of IPs:

$$\begin{aligned} \max \quad & x_2 \\ \text{subject to} \quad & 2x_1 - \frac{1}{h}x_2 \geq 0 \\ & 2x_1 + \frac{1}{h}x_2 \leq 2 \\ & x_1, x_2 \geq 0 \text{ and integer.} \end{aligned}$$

The feasible region of the continuous relaxation of this IP is an isosceles triangle with the base on the segment joining $(0, 0)$ and $(1, 0)$, and the top vertex at point $(\frac{1}{2}, h)$. The optimum of the continuous relaxation is the top vertex, and it has an objective value of h. The IP feasible region, however, only contains the two points $(0, 0)$ and $(1, 0)$ whose objective value is zero. Therefore, the difference between the objective value of the optimal solution of the continuous relaxation and the objective value of the IP optimal solution is $h - 0 = h$. This difference tends to infinity when $h \to \infty$.

Exercise 2

Drawing the feasible region of the continuous relaxation of the IP (see Figure 9), there are two potentially optimal vertices of the polytope. The first, say x_A, is at the intersection of the lines defining (11b) and (11c). The second, which we call x_B, is at the intersection of the lines defining (11b) and (11d). A quick computation yields:

$$x_A = \left(\frac{1}{2}, 3\right), \qquad z_A = \frac{66}{4};$$
$$x_B = \left(4, \frac{5}{4}\right), \qquad z_B = \frac{73}{4};$$

where z_A and z_B are the objective function values associated with x_A and x_B. We can conclude that $x^* = x_B$ is the optimum of the continuous relaxation of (11a)–(11e).

Current best integer solution:
None

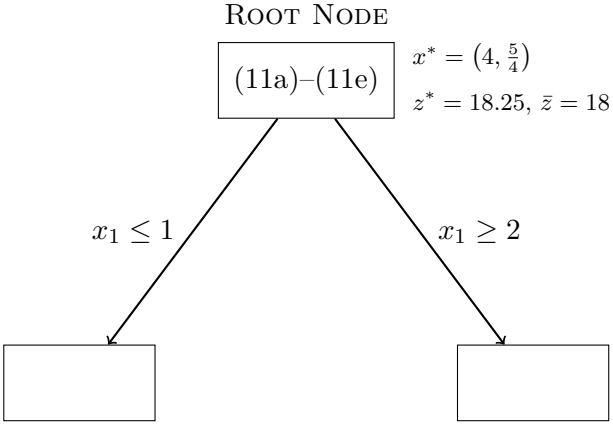

Figure 10: BB tree after solving the root node and branching.

Note that $\frac{73}{4}$ is an upper bound (i.e., a dual bound) on the optimum z^{opt} of the IP. However, because the coefficients appearing in the objective function are integers (indeed, they are 3 and 5) and, in any integer feasible solution, x_1 and x_2 are also integers, it follows that z^{opt} must be integer. Therefore, we can tighten this bound as follows:

$$z^{\text{opt}} \leq \left\lfloor \frac{73}{4} \right\rfloor = 18. \tag{12}$$

The only fractional coordinate of z^* is x_2. Therefore, we branch on $x_2 = 5/4$ and create two branches with constraints $x_2 \leq 1$ and $x_2 \geq 2$. Figure 10 displays the state of the BB tree up to this point.

Following the exercise instructions, we explore the left branch ($x_2 \leq 1$). (The feasible regions associated with the two branches are shown in Figure 11.) The optimum of the continuous relaxation in this branch is $x^* = (4, 1)$ with objective value $z^* = 17$. Because point x^* is integer feasible, its objective value is a lower bound for the optimum z^{opt}. Taking into account (12), we can already "squeeze" z^{opt} between an upper and a lower bound:

$$17 = \underline{z} \leq z^{\text{opt}} \leq \bar{z} = 18.$$

We update the BB tree as shown in Figure 12.

The only open node is the one associated with constraint $x_2 \geq 2$. Therefore, we will explore it. Potential optimal vertices are x_A (which we have already evaluated when solving the root node) with objective value $z_A = 66/4$ and $x_D = (5/2, 2)$ with objective value $70/4$. Therefore, x_D is the continuous relaxation optimum at this node. Furthermore, x_D is fractional (because its component x_1 is). However, remark that $66/4 = 16.5$. Therefore, the best possible value we can obtain at the subtree rooted at this node is 16.5. On the other hand, we have already encountered a strictly better integer solution $x_C = (4, 1)$ with objective value 17. We conclude that it is pointless to continue the exploration of this subtree, and we can prune it. As shown in Figure 13, no open node remains, and we have thus completed the execution of the BB algorithm. The optimal integer solution of (11a)–(11e) is $(4, 1)$.

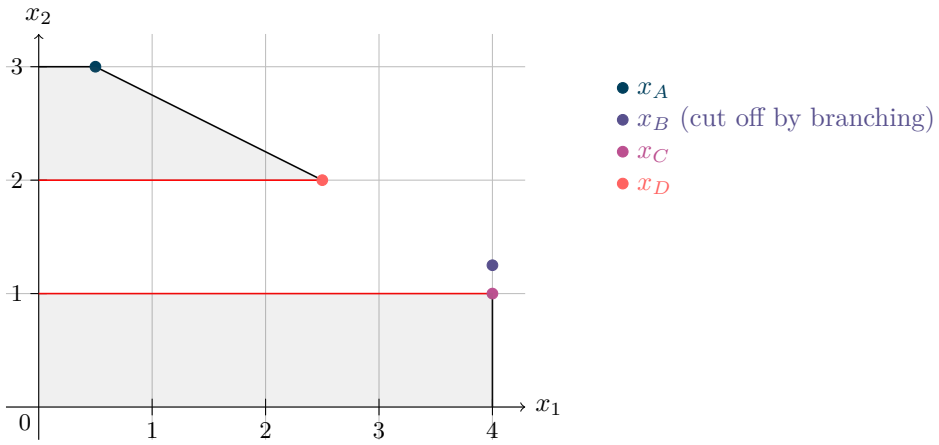

Figure 11: Feasible regions and relevant vertices of the subproblems obtained after the first branching.

Current best integer solution:
$x = (4, 1)$, $z = 17$

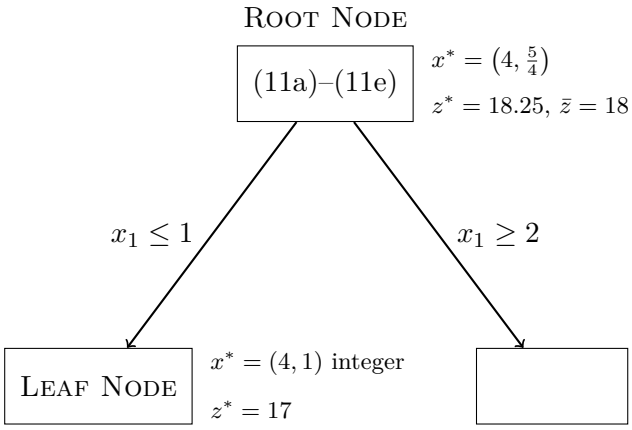

Figure 12: BB tree after solving the node associated with constraint $x_2 \leq 1$.

Current best integer solution:
$x = (4, 1)$, $z = 17$

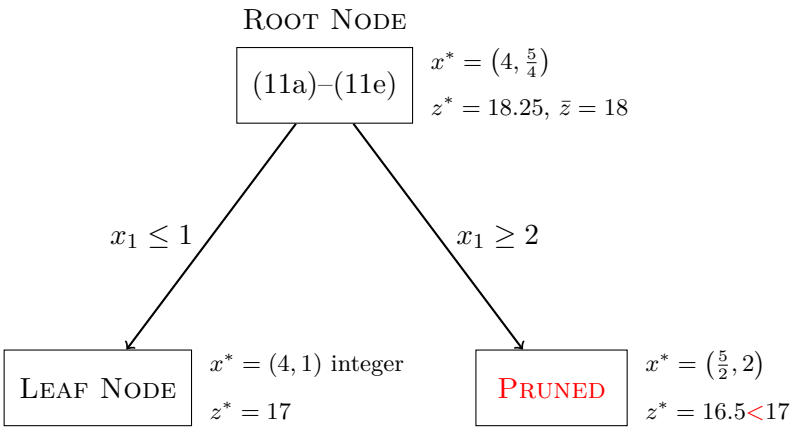

Figure 13: BB tree after completing the algorithm execution.