Combinatorial Optimization

Problem set 7: solutions

1. Formulate and solve an integer program for the following scenario.

> A trader of unusual objects is traveling with a caravan that begins in city A, proceeds through cities B, C, and D, in order, and ends in city E. The trader knows of some items in each city that can be purchased and later sold in other cities. The following table lists these items, their weights, their current locations, and the profit that can be gained by selling each item in later cities along the caravan route.

			Profit if sold in			
Item	Weight	In city	B	C	D	E
1	43	A	\$200	\$300	—	\$450
2	26	A	\$150	—	\$250	\$375
3	14	B		\$85	\$130	—
4	19	B		\$110	—	\$120
5	35	C			\$225	\$340
6	23	D				\$260

> The trader's camel can carry a maximum weight of 60. What items should the trader purchase, and where should the items be sold, in order to maximize profit by the end of the caravan route?

▷ **Solution.** For $1 \le i \le 6$ and $j \in \{\mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$, let $x_{ij} \in \{0, 1\}$ denote whether item i is to be sold in city j. Our objective is to maximize profit:

$$\begin{aligned}\text{maximize} \quad & 200x_{1\mathrm{B}} + 300x_{1\mathrm{C}} + 450x_{1\mathrm{E}} \\ & + 150x_{2\mathrm{B}} + 250x_{2\mathrm{D}} + 375x_{2\mathrm{E}} \\ & + 85x_{3\mathrm{C}} + 130x_{3\mathrm{D}} \\ & + 110x_{4\mathrm{C}} + 120x_{4\mathrm{E}} \\ & + 225x_{5\mathrm{D}} + 340x_{5\mathrm{E}} \\ & + 260x_{6\mathrm{E}}.\end{aligned}$$

The constraints must ensure that each item is sold at most once:

$$\begin{aligned} x_{1\mathrm{B}} + x_{1\mathrm{C}} + x_{1\mathrm{E}} &\le 1 \\ x_{2\mathrm{B}} + x_{2\mathrm{D}} + x_{2\mathrm{E}} &\le 1 \\ x_{3\mathrm{C}} + x_{3\mathrm{D}} &\le 1 \\ x_{4\mathrm{C}} + x_{4\mathrm{E}} &\le 1 \\ x_{5\mathrm{D}} + x_{5\mathrm{E}} &\le 1. \end{aligned}$$

Furthermore, we cannot exceed the camel's capacity on any leg of the journey:

$$\begin{aligned} 43(x_{1\mathrm{B}} + x_{1\mathrm{C}} + x_{1\mathrm{E}}) + 26(x_{2\mathrm{B}} + x_{2\mathrm{D}} + x_{2\mathrm{E}}) &\le 60 && [\mathrm{A} \to \mathrm{B}] \\ 43(x_{1\mathrm{C}} + x_{1\mathrm{E}}) + 26(x_{2\mathrm{D}} + x_{2\mathrm{E}}) + 14(x_{3\mathrm{C}} + x_{3\mathrm{D}}) + 19(x_{4\mathrm{C}} + x_{4\mathrm{E}}) &\le 60 && [\mathrm{B} \to \mathrm{C}] \\ 43x_{1\mathrm{E}} + 26(x_{2\mathrm{D}} + x_{2\mathrm{E}}) + 14x_{3\mathrm{D}} + 19x_{4\mathrm{E}} + 35(x_{5\mathrm{D}} + x_{5\mathrm{E}}) &\le 60 && [\mathrm{C} \to \mathrm{D}] \\ 43x_{1\mathrm{E}} + 26x_{2\mathrm{E}} + 19x_{4\mathrm{E}} + 35x_{5\mathrm{E}} + 23x_{6\mathrm{E}} &\le 60. && [\mathrm{D} \to \mathrm{E}] \end{aligned}$$

The variable domains are $x_{ij} \in \{0, 1\}$ for all $1 \le i \le 6$ and all $j \in \{\mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$.

The following Maple worksheet solves this integer program.

```
> restart;
> with(Optimization);
```

[*ImportMPS*, *Interactive*, *LPSolve*, *LSSolve*, *Maximize*, *Minimize*, *NLPSolve*, *QPSolve*]

```
> profit:=(x1b,x1c,x1e,x2b,x2d,x2e,x3c,x3d,x4c,x4e,x5d,x5e,x6e)->
    200*x1b+300*x1c+450*x1e+150*x2b+250*x2d+375*x2e+85*x3c+130*x3d
    +110*x4c+120*x4e+225*x5d+340*x5e+260*x6e;
```

$$profit := (x1b, x1c, x1e, x2b, x2d, x2e, x3c, x3d, x4c, x4e, x5d, x5e, x6e) \to 200\, x1b + 300\, x1c + 450\, x1e + 150\, x2b + 250\, x2d + 375\, x2e + 85\, x3c + 130\, x3d + 110\, x4c + 120\, x4e + 225\, x5d + 340\, x5e + 260\, x6e$$

```
> sellonce:=[x1b+x1c+x1e<=1,x2b+x2d+x2e<=1,x3c+x3d<=1,x4c+x4e<=1,
    x5d+x5e<=1];
```

$$sellonce := [x1b + x1c + x1e \le 1, x2b + x2d + x2e \le 1, x3c + x3d \le 1, x4c + x4e \le 1, x5d + x5e \le 1]$$

```
> capacity:=[43*(x1b+x1c+x1e)+26*(x2b+x2d+x2e)<=60,
    43*(x1c+x1e)+26*(x2d+x2e)+14*(x3c+x3d)+19*(x4c+x4e)<=60,
    43*x1e+26*(x2d+x2e)+14*x3d+19*x4e+35*(x5d+x5e)<=60,
    43*x1e+26*x2e+19*x4e+35*x5e+23*x6e<=60];
```

$$capacity := [43\, x1b + 43\, x1c + 43\, x1e + 26\, x2b + 26\, x2d + 26\, x2e \le 60, 43\, x1c + 43\, x1e + 26\, x2d + 26\, x2e + 14\, x3c + 14\, x3d + 19\, x4c + 19\, x4e \le 60, 43\, x1e + 26\, x2d + 26\, x2e + 14\, x3d + 19\, x4e + 35\, x5d + 35\, x5e \le 60, 43\, x1e + 26\, x2e + 19\, x4e + 35\, x5e + 23\, x6e \le 60]$$

```
> LPSolve(profit(x1b,x1c,x1e,x2b,x2d,x2e,x3c,x3d,x4c,x4e,x5d,x5e,
    x6e),[op(sellonce),op(capacity)],'maximize',assume=binary);
```

$$\big[1040, [x1b = 1, x1c = 0, x1e = 0, x2b = 0, x2d = 0, x2e = 0, x3c = 0, x3d = 1, x4c = 1, x4e = 0, x5d = 0, x5e = 1, x6e = 1]\big]$$

So the optimal strategy for the trader is to buy item 1 in city A; sell item 1 and buy items 3 and 4 in city B; sell item 4 and buy item 5 in city C; sell item 3 and buy item 6 in city D; and sell items 5 and 6 in city E. This will yield a total profit of \$1040. □

2. A *propositional formula* with the Boolean operators $\wedge$, $\vee$, and $\neg$, meaning AND, OR, and NOT, respectively, can be defined inductively as follows:

(i) A literal (i.e., a variable or its negation) is a propositional formula.

(ii) If P and Q are propositional formulas, then the conjunction $(P) \wedge (Q)$ is a propositional formula.

(iii) If P and Q are propositional formulas, then the disjunction $(P) \vee (Q)$ is a propositional formula.

(iv) If P is a propositional formula, then the negation $\neg(P)$ is a propositional formula.

Recall that a propositional formula is in *conjunctive normal form* (CNF) if it is a conjunction of disjunctions of literals. For example, the propositional formula

$$(x_1 \vee \bar{x}_2) \wedge (\bar{x}_2 \vee \bar{x}_3) \wedge (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_3)$$

is in conjunctive normal form.

(a) Prove that any propositional formula with the Boolean operators $\wedge$, $\vee$, and $\neg$ can be rewritten as an equivalent formula, on the same set of variables, in conjunctive normal form. (*Equivalent* means that the two formulas have the same set of satisfying variable assignments.)

(b) Suppose we introduce the Boolean operators $\oplus$ and $\rightarrow$, having the meanings "exclusive OR" and "implies," respectively. Show that any propositional formula with the Boolean operators $\wedge$, $\vee$, $\neg$, $\oplus$, and $\rightarrow$ can be rewritten as an equivalent formula in conjunctive normal form.

▷ **Solution.** Here is one approach. Suppose the variables in the given propositional formula F are $x_1, \ldots, x_n$. Construct a truth table for F, with one row for each of the 2^n possible assignments of truth values to the n variables.

For each assignment of truth values to the variables that makes F *false*, write a conjunction in which each variable appears positively if it is assigned the value TRUE and negatively if it is assigned the value FALSE. For example, for $n = 3$, if the assignment $x_1 =$ TRUE, $x_2 =$ FALSE, $x_3 =$ TRUE causes F to be false, then form the conjunction $x_1 \wedge \bar{x}_2 \wedge x_3$.

Next, negate each of these conjunctions, and then form the conjunction of the negations. The resulting formula disallows exactly those variable assignments that do not satisfy F. For instance, if the only two non-satisfying variable assignments are $x_1 =$ TRUE, $x_2 =$ FALSE, $x_3 =$ TRUE and $x_1 =$ FALSE, $x_2 =$ FALSE, $x_3 =$ FALSE then the resulting formula is

$$\neg(x_1 \wedge \bar{x}_2 \wedge x_3) \wedge \neg(\bar{x}_1 \wedge \bar{x}_2 \wedge \bar{x}_3).$$

Now apply De Morgan's law to each of the negations:

$$(\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge (x_1 \vee x_2 \vee x_3).$$

The resulting formula F' is in conjunctive normal form and is equivalent to F, because an assignment of truth values to the variables satisfies F' if and only if it does *not* cause F to be false.

Nothing in this construction depends on what Boolean operators appear in the formula F, so it works for both parts (a) and (b). □

3. Formulate and solve an integer program to determine truth values for the Boolean variables x_1, x_2, x_3, x_4, and x_5 so that the propositional formula

$$(x_1 \oplus x_2) \wedge (x_3 \vee x_4 \vee x_5) \wedge \neg[x_3 \wedge (x_4 \vee x_5)] \wedge (\bar{x}_4 \vee \bar{x}_5) \\ \wedge (x_1 \vee x_4 \vee x_5) \wedge (\bar{x}_1 \vee x_3 \vee x_5) \wedge (x_2 \vee x_4 \vee x_5) \wedge (\bar{x}_2 \vee x_3 \vee x_5)$$

is satisfied, or determine that the formula is unsatisfiable.

▷ **Solution.** First we convert the formula to conjunctive normal form. The subformula $x_1 \oplus x_2$ is equivalent to $(x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2)$, and the subformula $\neg[x_3 \wedge (x_4 \vee x_5)]$ is equivalent to

$$\neg[(x_3 \wedge x_4) \vee (x_3 \wedge x_5)] \equiv [\neg(x_3 \wedge x_4)] \wedge [\neg(x_3 \wedge x_5)] \equiv (\bar{x}_3 \vee \bar{x}_4) \wedge (\bar{x}_3 \vee \bar{x}_5)$$

by the distributive law and De Morgan's laws. So the given formula is equivalent to

$$(x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2) \wedge (x_3 \vee x_4 \vee x_5) \wedge (\bar{x}_3 \vee \bar{x}_4) \wedge (\bar{x}_3 \vee \bar{x}_5) \wedge (\bar{x}_4 \vee \bar{x}_5) \\ \wedge (x_1 \vee x_4 \vee x_5) \wedge (\bar{x}_1 \vee x_3 \vee x_5) \wedge (x_2 \vee x_4 \vee x_5) \wedge (\bar{x}_2 \vee x_3 \vee x_5).$$

An integer program to find a solution for this instance of the Boolean satisfiability problem appears below.

$$\begin{aligned} \text{maximize} \quad & 0 \\ \text{subject to} \quad & x_1 + x_2 \geq 1 \\ & (1 - x_1) + (1 - x_2) \geq 1 \\ & x_3 + x_4 + x_5 \geq 1 \\ & (1 - x_3) + (1 - x_4) \geq 1 \\ & (1 - x_4) + (1 - x_5) \geq 1 \\ & x_1 + x_4 + x_5 \geq 1 \\ & (1 - x_1) + x_3 + x_5 \geq 1 \\ & x_2 + x_4 + x_5 \geq 1 \\ & (1 - x_2) + x_3 + x_5 \geq 1 \\ & x_i \in \{0, 1\} \quad \text{for } 1 \leq i \leq 5. \end{aligned}$$

The following Maple worksheet solves this integer program.

```
> restart;
> with(Optimization);
```

[*ImportMPS*, *Interactive*, *LPSolve*, *LSSolve*, *Maximize*, *Minimize*, *NLPSolve*, *QPSolve*]

```
> clauses:=[x1+x2>=1,(1-x1)+(1-x2)>=1,x3+x4+x5>=1,(1-x3)+(1-x4)>=1,
    (1-x3)+(1-x5)>=1,(1-x4)+(1-x5)>=1,x1+x4+x5>=1,(1-x1)+x3+x5>=1,
    x2+x4+x5>=1,(1-x2)+x3+x5>=1];
```

$$clauses := [1 \leq x1 + x2, 0 \leq 1 - x1 - x2, 1 \leq x3 + x4 + x5, 0 \leq 1 - x3 - x4, \\ 0 \leq 1 - x3 - x5, 0 \leq 1 - x4 - x5, 1 \leq x1 + x4 + x5, 0 \leq -x1 + x3 + x5, \\ 1 \leq x2 + x4 + x5, 0 \leq -x2 + x3 + x5]$$

```
> LPSolve(0,clauses,assume=binary);
```

$$[0, [x1 = 0, x2 = 1, x3 = 0, x4 = 0, x5 = 1]]$$

So an assignment of truth values to variables that satisfies the given formula is $x_1 =$ FALSE, $x_2 =$ TRUE, $x_3 =$ FALSE, $x_4 =$ FALSE, $x_5 =$ TRUE. □

[The other satisfying assignment is $x_1 =$ TRUE, $x_2 =$ FALSE, $x_3 =$ FALSE, $x_4 =$ FALSE, $x_5 =$ TRUE.]

4. Is there a maximum possible (Euclidean) distance between the optimal solution of an integer program and the optimal solution of its LP relaxation? If so, give an upper bound for this distance and prove that it is an upper bound. If not, show how to construct, given any positive real number M, an integer program whose optimal solution is at least distance M from the optimal solution of its LP relaxation.

▷ **Solution.** No, there is no maximum possible distance between the optimal solution of an integer program and the optimal solution of its LP relaxation.

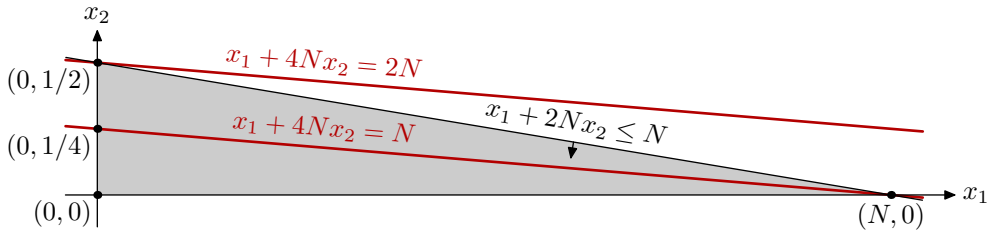
Here is one construction. Let $M > 0$. Let N be an integer greater than or equal to M (for example, $N = \lceil M \rceil$). Consider the integer program

$$\begin{aligned} \text{maximize} \quad & x_1 + 4Nx_2 \\ \text{subject to} \quad & x_1 + 2Nx_2 \leq N \\ & x_1 \geq 0, \quad x_2 \geq 0 \\ & x_1, x_2 \text{ integer.} \end{aligned}$$

The LP relaxation is

$$\begin{aligned} \text{maximize} \quad & x_1 + 4Nx_2 \\ \text{subject to} \quad & x_1 + 2Nx_2 \leq N \\ & x_1 \geq 0, \quad x_2 \geq 0. \end{aligned}$$

The feasible region of the LP relaxation is shaded in the figure below. Note that the corners of the feasible region are $(0, 0)$, $(N, 0)$, and $(0, 1/2)$. Two level curves are drawn: $x_1 + 4Nx_2 = N$ and $x_1 + 4Nx_2 = 2N$.



The (unique) optimal solution to the LP relaxation is $x_1 = 0$, $x_2 = 1/2$, with objective value $2N$. This can be seen to be optimal by multiplying the constraint by 2:

$$2x_1 + 4Nx_2 \leq 2N.$$

So, since x_1 and x_2 are nonnegative, this implies

$$x_1 + 4Nx_2 \leq 2x_1 + 4Nx_2 \leq 2N.$$

Hence a solution that achieves the objective value $2N$ must be optimal.

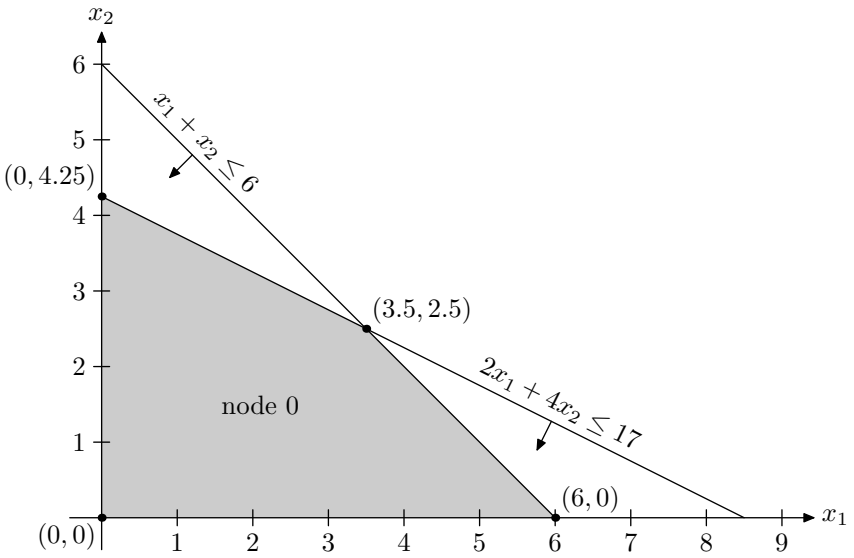
Note that the domain $x_1 \geq 0$ and the constraint $x_1 + 2Nx_2 \leq N$ together imply $2Nx_2 \leq N$, that is, $x_2 \leq 1/2$. So all feasible integer solutions must have $x_2 = 0$. Therefore the optimal integer solution is $x_1 = N$, $x_2 = 0$, having objective value N.

The x_1-coordinates of these two solutions differ by $N \geq M$, so certainly their distance apart is at least M. □

5. Solve the following integer program using the branch-and-bound technique.

$$\begin{aligned} \text{maximize} \quad & 2x_1 + 3x_2 \\ \text{subject to} \quad & x_1 + x_2 \leq 6 \\ & 2x_1 + 4x_2 \leq 17 \\ & x_1 \geq 0, \quad x_2 \geq 0 \\ & x_1,\, x_2 \text{ integer.} \end{aligned}$$

▷ **Solution.** The feasible region of the LP relaxation appears below.



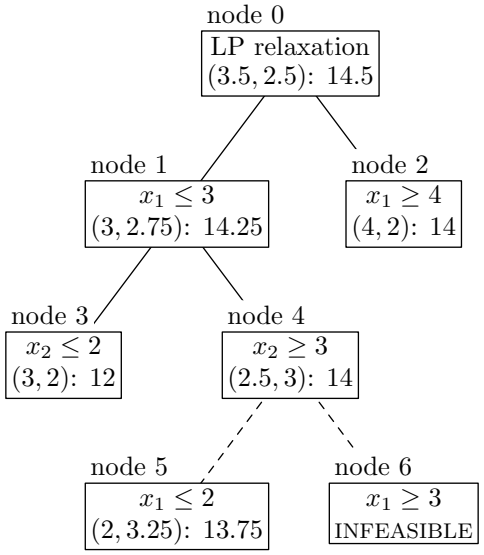
This will correspond to node 0 of our branch-and-bound tree. We solve the LP relaxation by evaluating the objective function at the corners of the feasible region.

Corner	Obj. value
$(0, 0)$	0
$(6, 0)$	12
$(3.5, 2.5)$	14.5
$(0, 4.25)$	12.75

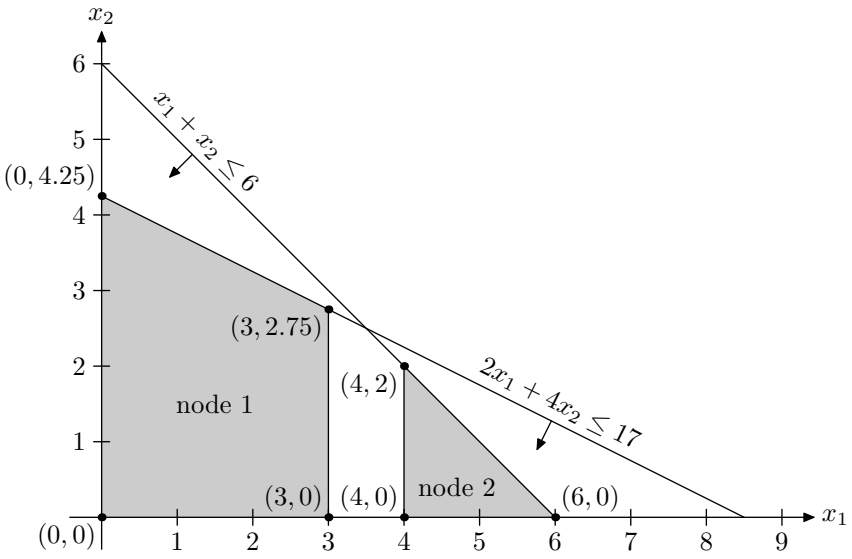
So the optimal solution to the LP relaxation is $x_1 = 3.5$, $x_2 = 2.5$, with objective value 14.5.

Unfortunately, this is not a feasible solution to the integer program, because neither x_1 nor x_2 has an integer value. So we will choose a variable with a non-integer value and branch on it.

If we choose to branch on x_1 first, we will build the following branch-and-bound tree. The nodes are numbered in the order they are created. The top line of each node label specifies the constraint that has been added to the LP relaxation (in addition to the constraints for nodes higher up in the tree), and the bottom line gives the optimal solution and optimal objective value for the corresponding linear program with the added constraints.



We begin by excluding the fractional value $x_1 = 3.5$ by adding one of two new constraints to the LP relaxation: $x_1 \leq 3$ or $x_1 \geq 4$. Any feasible integer solution must satisfy one or the other of these two constraints. This gives us two new LPs to solve, represented by nodes 1 and 2 in the branch-and-bound tree above. The feasible regions for these LPs are shown below.

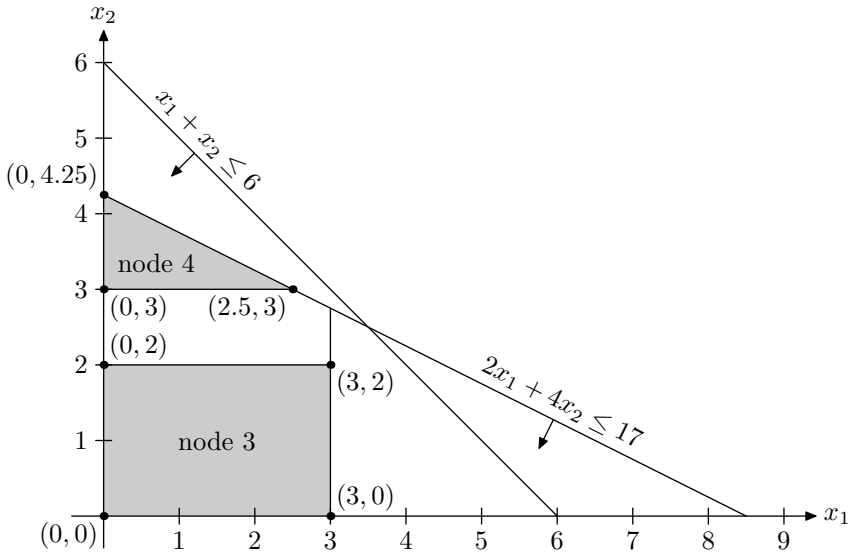


The objective function has already been evaluated at some of the corners in this picture; the objective values at the new corners are given in the table below.

Corner	Obj. value
$(3, 0)$	6
$(3, 2.75)$	14.25
$(4, 0)$	8
$(4, 2)$	14

So the optimal solution to the LP for node 1 is $x_1 = 3$, $x_2 = 2.75$, with objective value 14.25, and the optimal solution to the LP for node 2 is $x_1 = 4$, $x_2 = 2$, with objective value 14.

Now, node 1 has the larger optimal objective value, but in the optimal solution the value of x_2 is not an integer. So we must branch on x_2 next. We exclude the fractional value $x_2 = 2.75$ by adding one of two new constraints: $x_2 \leq 2$ or $x_2 \geq 3$. This gives us two new LPs, represented by nodes 3 and 4 in the branch-and-bound tree. The feasible regions for these LPs are shown below.



The objective values at the new corners are given in the table below.

Corner	Obj. value
$(3, 2)$	12
$(0, 2)$	6
$(2.5, 3)$	14
$(0, 3)$	9

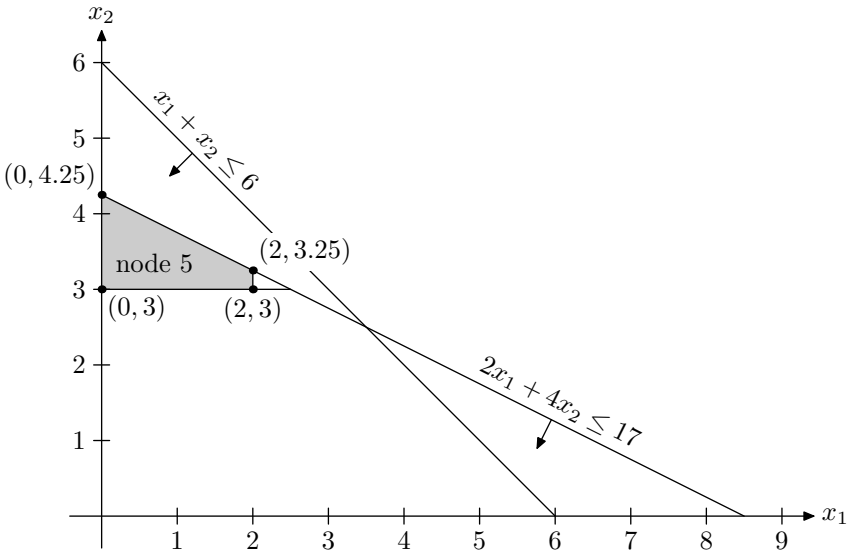
So the optimal solution to the LP for node 3 is $x_1 = 3$, $x_2 = 2$, with objective value 12, and the optimal solution to the LP for node 4 is $x_1 = 2.5$, $x_2 = 3$, with objective value 14.

Now, at this point we have a feasible *integer* solution with objective value 14 (the optimal solution to the LP for node 2), and at the other leaf nodes (nodes 3 and 4) the optimal objective values for the LPs are no greater than 14, so we can conclude that the solution $x_1 = 4$, $x_2 = 2$ found at node 2 is *optimal*, because there cannot be a better solution anywhere else in the branch-and-bound tree.

[In fact, with slightly more clever reasoning, we could have stopped earlier: Because the objective function coefficients are integers, the objective value of any integer solution must be an integer. Therefore, the bound of 14.25 at node 1 really means that the best possible objective value for an integer solution below that node is $\lfloor 14.25 \rfloor = 14$, so we cannot possibly do *better* than the objective value 14 achieved by the integer solution at node 2. So we didn't really need to search below node 1. And actually, we knew at the very beginning (at node 0) that no integer solution could have objective value better than 14, because the optimal objective value of the LP relaxation was 14.5, so the objective value of any integer solution can be no greater than $\lfloor 14.5 \rfloor = 14$.]

If we want to make sure that we have *all* optimal solutions to the integer program, we must continue exploring below node 4, because there may be another integer solution with objective value as good as 14 in that part of the tree. So we explore below node 4 by branching on x_1. We exclude the fractional value $x_1 = 2.5$ by adding one of two new constraints: $x_1 \leq 2$ or $x_1 \geq 3$. This gives us two new LPs, represented by nodes 5 and 6 in

the branch-and-bound tree. The feasible regions of these new LPs are shown below. Note that the constraint $x_1 \geq 3$, in conjunction with the constraints added in higher nodes in the branch-and-bound tree, makes the LP infeasible (because no part of the feasible region for node 4 has $x_1 \geq 3$).

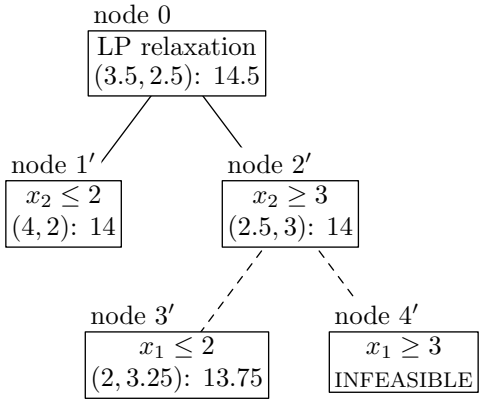


The objective values at the new corners are given in the table below.

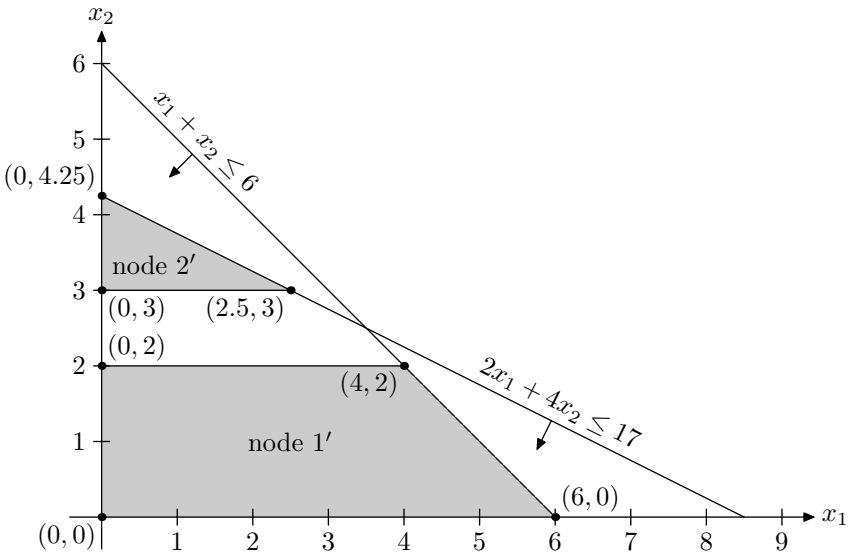
Corner	Obj. value
$(2, 3)$	13
$(2, 3.25)$	13.75

So the optimal solution to the LP for node 5 is $x_1 = 2$, $x_2 = 3.25$, with objective value 13.75. Now we know that the integer solution found at node 2 is the unique optimal integer solution, because all of the other leaves in the branch-and-bound tree (nodes 3, 5, and 6) either have strictly smaller bounds or are infeasible.

Alternatively, at the beginning, after node 0, we could have chosen to branch on x_2 rather than x_1. In that case, we would have built the following branch-and-bound tree.



In this case, we begin by excluding the fractional value $x_2 = 2.5$ by adding one of two new constraints: $x_2 \leq 2$ or $x_2 \geq 3$. This gives us two new LPs, represented by nodes $1'$ and $2'$ in the branch-and-bound tree. The feasible regions of these new LPs are shown below.



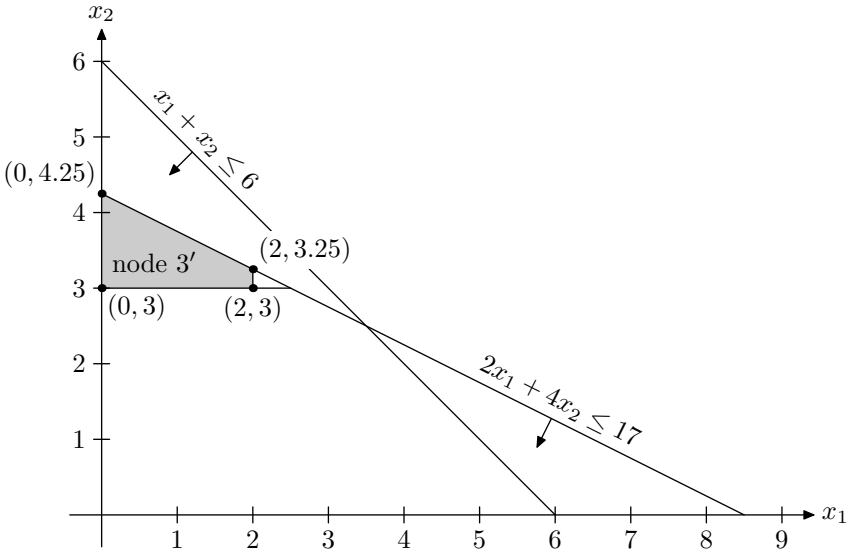
The objective values of the corners not evaluated for node 0 are given in the table below.

Corner	Obj. value
$(4, 2)$	14
$(0, 2)$	6
$(2.5, 3)$	14
$(0, 3)$	9

So the optimal solution to the LP for node $1'$ is $x_1 = 4$, $x_2 = 2$, with objective value 14, and the optimal solution for the LP for node $2'$ is $x_1 = 2.5$, $x_2 = 3$, also with objective value 14.

At node $1'$ we have an integer solution with objective value 14, and at the other leaf node, node $2'$, we have a bound of 14, meaning that no solution below that node in the tree can have an objective value better than 14. Hence the solution $x_1 = 4$, $x_2 = 2$ is optimal.

However, as before, if we want to make sure we have *all* optimal integer solutions, then we can continue to explore below node $2'$. We branch on x_1, excluding the fractional value $x_1 = 2.5$ by adding one of two new constraints: $x_1 \leq 2$ or $x_1 \geq 3$. This gives us two new LPs, represented by the nodes $3'$ and $4'$ in the branch-and-bound tree above. The feasible regions of these new LPs are shown below. Note that the LP for node $4'$ is infeasible.



The objective values at the new corners are given in the table below.

Corner	Obj. value
$(2, 3)$	13
$(2, 3.25)$	13.75

So the optimal solution to the LP for node $3'$ is $x_1 = 2$, $x_2 = 3.25$. Now we know that the integer solution found at node $1'$ is the unique optimal integer solution, because all of the other leaves in the branch-and-bound tree (nodes $3'$ and $4'$) either have strictly smaller bounds or are infeasible. □

6. In the CUBIC SUBGRAPH problem, the input is a graph $G = (V, E)$, and the question to be answered is whether there exists a subgraph $H = (V', E')$ of G that is *cubic*, meaning that every vertex in the graph H has degree 3. Describe how to formulate an integer program to solve this problem, given a graph $G = (V, E)$. Justify that your formulation correctly solves the problem.

▷ **Solution.** For each vertex $v \in V$, let $x_v \in \{0, 1\}$ indicate whether $v \in V'$. For each edge $e \in E$, let $y_e \in \{0, 1\}$ indicate whether $e \in E'$. For a vertex $v \in V$, let $E_v = \{\, e \in E : v \in e \,\}$ denote the set of edges incident upon v. The following integer program can be used to solve this problem.

$$\begin{aligned}
\text{maximize} \quad & 0 \\
\text{subject to} \quad & \sum_{v \in V} x_v \geq 1 \\
& \sum_{e \in E_v} y_e = 3x_v \qquad \text{for all } v \in V \\
& x_v \in \{0, 1\} \qquad \text{for all } v \in V \\
& y_e \in \{0, 1\} \qquad \text{for all } e \in E.
\end{aligned}$$

The objective function is 0 because we are not minimizing or maximizing anything; we are merely searching for a feasible solution (which corresponds to a cubic subgraph of G). The first constraint ensures that not all of the variables x_v will be 0, so that V' will be nonempty. The remaining constraints ensure that if $x_v = 1$ then exactly three edges incident upon v are chosen to be in E' (so v has degree 3 in H), and if $x_v = 0$ then no edges incident upon v are chosen to be in E'. Therefore every feasible solution to this integer program corresponds to a cubic subgraph of G, and vice versa, so this integer program will be feasible if and only if G has a cubic subgraph. □