Assignment 5a: Linear Programming

Pietro Bonazzi

Economics and Computation, MSc Course, 1st Sem., Fall 2020
University of Zürich
pietro.bonazzi@uzh.ch

Contents

Chapter 1

LP or Not?

For each of the following examples, determine whether it is a linear programming problem. If not, explain why.

1.1 a

Problem 1 is a linear programming problem because all the four assumptions for a linear programming problem are full filled.

$$\begin{array}{ll} \text{maximize} & 2x_1 + 7x_2 \\ \text{subject to} & 2x_1 + 6x_2 \leq 11 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{array} \tag{1}$$

Figure 1.1.1: Problem 1.

1.2 b

Problem 2 violates the assumption of additivity in the first term of the first constraint. Therefore, it is not a linear programming problem.

$$\begin{array}{ll} \text{minimize} & 4x_1 + 6x_2 + 4x_1 \\ \text{subject to} & 23x_1x_2 + 5x_2 \leq 11 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{array} \tag{2}$$

Figure 1.2.1: Problem 2.

1.3 c

Problem 3 violates the assumption of proportionality in its objective function. Therefore, it is not a linear programming problem.

$$\begin{array}{ll} \text{maximize} & 9{x_1}^2 + 6x_1x_2 + {x_2}^2 \\ \text{subject to} & 27x_1 + 6x_1 \leq 25 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{array} \tag{3}$$

Figure 1.3.1: Problem 3.

1.4 d

Problem 4 is a linear programming problem because all the four assumptions for a linear programming problem are full filled. However, no feasible solutions exist for this problem because the first constraint cannot be full filled with the other two.

$$\begin{array}{ll} \text{maximize} & 65x_1 + 128x_2 \\ \text{subject to} & x_1 + x_2 \geq 31 \\ & x_1 \leq 0 \\ & x_2 \leq 0 \end{array} \tag{4}$$

Figure 1.4.1: Problem 4.

1.5 e

Problem 5 is not linear programming problem because it cannot be asked that a variable assumes *two different fixed values* at the same time.

$$\begin{array}{ll} \text{maximize} & 3x_1 - 5x_2 \\ \text{subject to} & 2x_1 + x_2 \leq 15 \\ & x_1 + 4x_2 \geq 10 \\ & x_1 = 0 \\ & x_1 = 1 \end{array} \tag{5}$$

Figure 1.5.1: Problem 5.

Chapter 2

Java/JOpt warm up.

2.1 Problem 6.

The implementation of the Problem in Figure 2.2.1 is detailed in the box code below.

$$
\begin{array}{ll}
\text{maximize} & 0.63x_1 + 0.37x_2 \\
\text{subject to} & x_1 - x_2 \geq 2 \\
 & x_1 + 0.25x_2 = 3 \\
 & x_1 \geq 0 \\
 & x_2 \geq 0
\end{array} \tag{6}
$$

Figure 2.1.1: Problem 6.

```java
        Variable x = new Variable("x", VarType.DOUBLE, -MIP.MAX_VALUE, MIP.MAX_VALUE);
        Variable y = new Variable("y", VarType.DOUBLE, -MIP.MAX_VALUE, MIP.MAX_VALUE);

        linearProgram.add(x);
        linearProgram.add(y);

        linearProgram.setObjectiveMax(true);
        linearProgram.addObjectiveTerm(0.63, x);
        linearProgram.addObjectiveTerm(0.37, y);

        Constraint c1 = new Constraint(CompareType.GEQ, 2);
        c1.addTerm(1, x);
        c1.addTerm(-1, y);
        linearProgram.add(c1);

        Constraint c2 = new Constraint(CompareType.EQ, 3);
        c2.addTerm(1, x);
        c2.addTerm(0.25, y);
        linearProgram.add(c2);

        Constraint c3 = new Constraint(CompareType.GEQ, 0);
        c3.addTerm(1, x);
        linearProgram.add(c3);

        Constraint c4 = new Constraint(CompareType.GEQ, 0);
        c4.addTerm(1, y);
        linearProgram.add(c4);
```

The values of the variables in the objective function are x1 (or x) : 2.8000000000000003 , x2 (or y) : 0.8000000000000003

The optimal solution of the objective function is 2.0600000000000005

2.2 Problem 7.

The implementation of the Problem in Figure 2.2.1 is detailed in the box code below.

$$
\begin{array}{ll}
\text{maximize} & 4x_1 + 2x_2 \\
\text{subject to} & 2x_1 + x_2 \leq 3 \\
 & x_1 + 2x_2 \leq 3 \\
 & x_1 \geq 0 \\
 & x_2 \geq 0
\end{array} \tag{7}
$$

Figure 2.2.1: Problem 7.

```
        Variable x = new Variable("x", VarType.DOUBLE, -MIP.MAX_VALUE, MIP.MAX_VALUE);
        Variable y = new Variable("y", VarType.DOUBLE, -MIP.MAX_VALUE, MIP.MAX_VALUE);

        linearProgram.add(x);
        linearProgram.add(y);

        linearProgram.setObjectiveMax(true);
        linearProgram.addObjectiveTerm(4, x);
        linearProgram.addObjectiveTerm(2, y);

        Constraint c1 = new Constraint(CompareType.LEQ, 3);
        c1.addTerm(2, x);
        c1.addTerm(1, y);
        linearProgram.add(c1);

        Constraint c2 = new Constraint(CompareType.LEQ, 3);
        c2.addTerm(1, x);
        c2.addTerm(2, y);
        linearProgram.add(c2);

        Constraint c3 = new Constraint(CompareType.GEQ, 0);
        c3.addTerm(1, x);
        linearProgram.add(c3);

        Constraint c4 = new Constraint(CompareType.GEQ, 0);
        c4.addTerm(1, y);
        linearProgram.add(c4);
```

The values of the variables in the objective function are x1 (or x) : 1.0 , x2 (or y) : 1.0 The optimal solution of the objective function is 6.0

2.3 Multiple solutions.

No. There are multiple solutions possible. These solutions are found in the grey area in Figure 2.3.1.

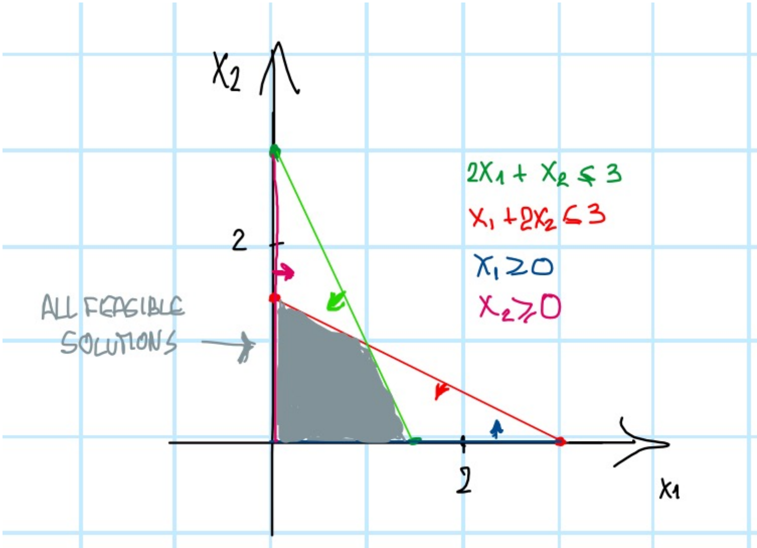

Figure 2.3.1: Multiple options.

Chapter 3

Local Lemonade Ltd.

3.1 Formulate the problem.

The problem is formulated as follows, see 3.1.1.

Z = profit

X_1 = nº sweet lemonade

X_2 = nº diet lemonade

C1 = 30 − 8 − 6 − 6 = 10

C2 = 20 − 6 − 2 − 6 = 6

Maximize $Z = \overset{C1}{10}X_1 + \overset{C2}{6}X_2$ objective Function

$6X_1 + 6X_2 \leq 2\,800$ constraint in Kg sugar available

$6X_1 + 2X_2 \leq 3000$ constraint in Kg lemon available

$X_1 \geq 0$ and $X_2 \geq 0$ nonnegativity constraints

Figure 3.1.1: Formulation of the problem.

The objective function maximizes the company profits. In normal conditions, the company makes 10 dollar for each hectolitre of sweet lemonade and 6 dollar for the diet lemonade. The number of items cannot be negative and there are two constraints in the kilos of sugar and lemon available for production.

3.2 Implement the linear program in Java/JOpt.

The implementation of the Problem is detailed in the box code below.

```
Variable x = new Variable("x", VarType.DOUBLE, -MIP.MAX_VALUE, MIP.MAX_VALUE);
Variable y = new Variable("y", VarType.DOUBLE, -MIP.MAX_VALUE, MIP.MAX_VALUE);

linearProgram.add(x);
linearProgram.add(y);

linearProgram.setObjectiveMax(true);
linearProgram.addObjectiveTerm(10, x);
linearProgram.addObjectiveTerm(6, y);

Constraint c1 = new Constraint(CompareType.LEQ, 2800);
c1.addTerm(6, x);
c1.addTerm(6, y);
linearProgram.add(c1);

Constraint c2 = new Constraint(CompareType.LEQ, 3000);
c2.addTerm(6, x);
c2.addTerm(2, y);
linearProgram.add(c2);

Constraint c3 = new Constraint(CompareType.GEQ, 0);
c3.addTerm(1, x);
linearProgram.add(c3);

Constraint c4 = new Constraint(CompareType.GEQ, 0);
c4.addTerm(1, y);
linearProgram.add(c4);
```

To maximize its profit, the operation department at Local Lemonade Ltd should produce 466 hectolitres of Sweet Lemonade and 0 hectolitre of Diet Lemonade. The total profit will be around CHF 4666.

3.3 New linear programming problem: Accident.

The problem is formulated as follows, see Figure 3.3.1.

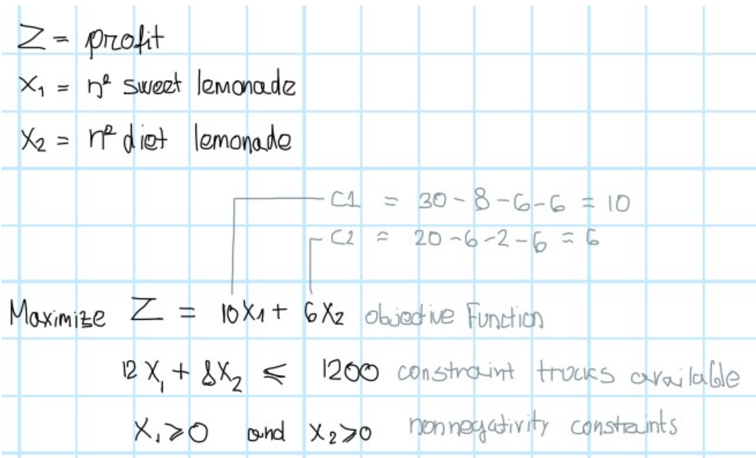

Figure 3.3.1: Formulation of the problem.

The accidents causes limitations in the kilos of raw material available for production. This Problem is implemented as follows :

```
        Variable x = new Variable("x", VarType.DOUBLE, -MIP.MAX_VALUE, MIP.MAX_VALUE);
        Variable y = new Variable("y", VarType.DOUBLE, -MIP.MAX_VALUE, MIP.MAX_VALUE);

        linearProgram.add(x);
        linearProgram.add(y);

        linearProgram.setObjectiveMax(true);
        linearProgram.addObjectiveTerm(10, x);
        linearProgram.addObjectiveTerm(6, y);

        Constraint c1 = new Constraint(CompareType.LEQ, 1200);
        c1.addTerm(12, x);
        c1.addTerm(8, y);
        linearProgram.add(c1);

        Constraint c2 = new Constraint(CompareType.GEQ, 0);
        c2.addTerm(1, x);
        linearProgram.add(c2);

        Constraint c3 = new Constraint(CompareType.GEQ, 0);
        c3.addTerm(1, y);
        linearProgram.add(c3);
```

The accident decreases the maximum profit that can be obtained. The total profit will be around CHF 1000. To reach this value, only 100 hectolitres of Sweet Lemonade should be produced.

3.4 New linear programming problem: Small back road.

The problem is formulated as follows, see Figure 3.4.1.

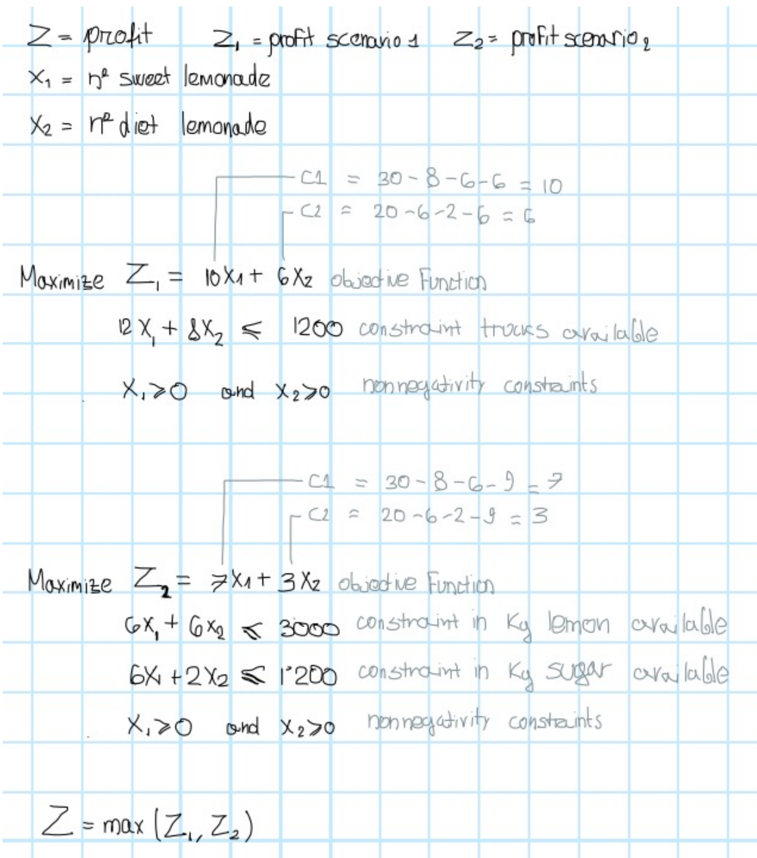

Figure 3.4.1: Formulation of the problem.

The implementation of the first linear problem can be found in Exercise 3.3, whereas the second part of the Problem is implemented as follows :

```
        Variable x = new Variable("x", VarType.DOUBLE, -MIP.MAX_VALUE, MIP.MAX_VALUE);
        Variable y = new Variable("y", VarType.DOUBLE, -MIP.MAX_VALUE, MIP.MAX_VALUE);

        linearProgram.add(x);
        linearProgram.add(y);

        linearProgram.setObjectiveMax(true);
        linearProgram.addObjectiveTerm(7, x);
        linearProgram.addObjectiveTerm(3, y);

        Constraint c1 = new Constraint(CompareType.LEQ, 3000);
        c1.addTerm(6, x);
        c1.addTerm(6, y);
        linearProgram.add(c1);

        Constraint c2 = new Constraint(CompareType.LEQ, 1200);
        c2.addTerm(6, x);
        c2.addTerm(2, y);
        linearProgram.add(c2);

        Constraint c3 = new Constraint(CompareType.GEQ, 0);
        c3.addTerm(1, x);
        linearProgram.add(c3);

        Constraint c4 = new Constraint(CompareType.GEQ, 0);
        c4.addTerm(1, y);
        linearProgram.add(c4);
```

Local Lemonade Ltd's finances are better off if its executives accept the offer of the supplier of lemon. The maximum profit that can be obtained becomes CHF 1700. To reach this value, the Local Lemonade Ltd should produce 50 hectolitres of Sweet Lemonade and 450 hectolitres of Diet Lemonade.