

Lecture 1 — Writing a Linear Program

From a problem in words to a mathematical model



Dr. Yahia Djemmada

National Higher School of Mathematics (NHSM), Algiers

yahia.djemmada@nhsm.edu.dz

Operations Research (O.R.)

What you will learn. After this lecture you can read a problem described in words and write it as a *linear program*: you choose the unknowns, you write what you want to maximise or minimise, and you write the limits as equations or inequalities. You will also learn to write the same program in different forms. Solving linear programs is the subject of Lecture 2.

How to read this lecture. Each idea starts with an example with numbers. Definitions, rules and methods are in boxes, and each rule comes with the reason why it is true. The “Check yourself” questions have their answers at the end of the lecture.

Contents

1	What is operations research?	2
2	A first example, step by step: the drone workshop	3
2.1	The problem	3
2.2	Question 1: what do we decide? The variables	3
2.3	Question 2: what do we want? The objective	3
2.4	Question 3: what limits us? The constraints	4
2.5	The complete model	4
2.6	Vocabulary	4
2.7	Whole numbers: lucky or not?	5
3	What does “linear” mean?	6
4	A method in five steps	6
5	Example 1.2: mixing raw materials	7
6	Example 1.3: transport	8
7	Example 1.4: assignment	9
8	Writing a linear program with matrices	9
8.1	Why do we need matrices?	10

8.2	Columns, lines, and the product of a line by a column	10
8.3	A matrix times a column	10
8.4	Comparing two columns	11
8.5	The drone workshop in matrix form	11
8.6	Any linear program, with sums and with matrices	12
8.7	Constraints with different signs	12
9	Different forms of the same linear program	12
9.1	Why several forms?	12
9.2	Five rules to change the form	13
9.3	The rules in practice	14
9.4	Worked example (a): the drone workshop in standard form	15
9.5	Worked example (b): from a general linear program to the canonical form	15
10	Useful tricks, and what is impossible	16
10.1	Absolute values	16
10.2	The smallest or the largest of several values	16
10.3	What is impossible, and how to recognise it	17
11	What to do with a strict inequality	17
11.1	The problem	17
11.2	Case 1: the quantities are whole numbers	17
11.3	Case 2: the quantities are real numbers	18
11.4	An example with real numbers	19
12	The four hypotheses of linear programming	19
13	Summary	20
	Answers to “Check yourself”	20
	Appendix: the best plans of the four examples	21

1 What is operations research?

Operations research (OR) uses mathematics to make the *best decision* when resources are *limited*: time, money, material, energy. The work always has two steps:

1. write a *model*: translate the problem into mathematics;
2. *solve* the model, usually with an algorithm.

Linear programming (LP) is the most used model of operations research. In a linear program, all expressions are *linear*: numbers multiplied by unknowns, then added, like $40x_1 + 60x_2$.

A little history: L. Kantorovich wrote the first linear programs in 1939 to plan production; G. Dantzig invented the simplex method, the main algorithm to solve them, in 1947; Kantorovich and T. Koop-

mans received the Nobel Prize in Economics in 1975 for this work.

Linear programming is used in factories (how much to produce), in industry (how to mix raw materials), in logistics (which warehouse supplies which customer), for planning over time (production, stocks, stored energy) and for schedules. Companies that use robots or drones meet the same questions: which robot does which task, where to send spare parts, when to charge the batteries.

2 A first example, step by step: the drone workshop

2.1 The problem

A small workshop assembles two models of drones: model A (simple) and model B (with a camera). The data are:

	Model A	Model B	Available per week
Assembly (hours for one drone)	1	2	40 hours
Testing (hours for one drone)	3	2	60 hours
Cameras (for one drone)	0	1	18 cameras
Profit (thousand DA for one drone)	40	60	

Question: how many drones of each model should be made each week to get the largest profit?

To write the model we answer three questions: *What do we decide? What do we want? What limits us?*

2.2 Question 1: what do we decide? The variables

We must decide two numbers. We give them names:

- ▶ x_1 = number of drones A made per week;
- ▶ x_2 = number of drones B made per week.

These unknowns are the *decision variables*. Drones are counted in whole numbers; we explain in Section 2.5 how the model treats this. A choice of values is a *plan*: for example $(x_1, x_2) = (10, 5)$ means “make 10 drones A and 5 drones B each week”.

Important habit. Always write in a full sentence what each variable means, with its unit (number of drones, kg, hours, ...) and its period (per day, per week, ...).

2.3 Question 2: what do we want? The objective

One drone A gives 40 thousand DA (Algerian dinars), so x_1 drones A give $40x_1$ thousand DA. One drone B gives 60 thousand DA, so x_2 drones B give $60x_2$ thousand DA. We count the profit in thousands of dinars to keep small numbers. The total profit is

$$z = 40x_1 + 60x_2 \quad (\text{to maximise}).$$

Example. For the plan $(10, 5)$: $z = 40 \times 10 + 60 \times 5 = 400 + 300 = 700$ thousand DA, that is 700 000 DA.

Units. (thousand DA per drone) $\times$ (number of drones) = thousand DA.

The function z is the *objective function*. We *maximise* it when it is a profit or a gain; we *minimise* it when it is a cost, a time or an energy.

2.4 Question 3: what limits us? The constraints

- **Assembly.** One drone A needs 1 hour, so x_1 drones A need $1 \times x_1 = x_1$ hours. One drone B needs 2 hours, so x_2 drones B need $2x_2$ hours. In total $x_1 + 2x_2$ hours, and we have at most 40 hours:

$$x_1 + 2x_2 \leq 40.$$

- **Testing.** In the same way: $3x_1 + 2x_2 \leq 60$.
- **Cameras.** Each drone B needs one camera, drones A need none, and we have at most 18 cameras: $x_2 \leq 18$.
- **Signs.** We cannot make a negative number of drones: $x_1 \geq 0$ and $x_2 \geq 0$. These are the *sign constraints*. Never forget them.

2.5 The complete model

$$\begin{array}{ll} \max & z = 40x_1 + 60x_2 \quad (\text{profit, thousand DA per week}) \\ \text{subject to} & x_1 + 2x_2 \leq 40 \quad (\text{assembly hours}) \\ & 3x_1 + 2x_2 \leq 60 \quad (\text{testing hours}) \\ & x_2 \leq 18 \quad (\text{cameras}) \\ & x_1 \geq 0, x_2 \geq 0 \quad (\text{signs}) \end{array}$$

“Subject to” is often written “s.t.”.

Whole numbers or real numbers?

Drones are whole objects. So we must distinguish two problems:

- the *real problem*: the same objective and constraints, with $x_1, x_2 \in \{0, 1, 2, \dots\}$ (whole numbers). A problem with whole-number variables is called an *integer program*;
- its *LP relaxation*: the linear program above, where x_1, x_2 can be any real numbers ≥ 0 .

In this course we solve the LP relaxation, because it is much easier. Section 2.7 explains when its answer is also the answer of the real problem.

Definition 1.1 — Linear program

A *linear program* (LP) is a problem where we maximise or minimise a linear function of the variables, subject to linear constraints. Each constraint is a linear expression which is “ $\leq$ ”, “ $=$ ” or “ $\geq$ ” a number.

2.6 Vocabulary

Definition 1.2 — Feasible and optimal solutions

- A *feasible solution* is a plan that satisfies *all* the constraints (including the sign constraints).
- The *feasible set* is the set of all feasible solutions.
- An *optimal solution* is a feasible solution with the best value of the objective (the largest for a maximisation, the smallest for a minimisation). This best value is the *optimal value*.

Example 1. Is (10, 10) feasible? Assembly: $10 + 2 \times 10 = 30 \leq 40$, yes. Testing: $3 \times 10 + 2 \times 10 = 50 \leq 60$, yes. Cameras: $10 \leq 18$, yes. Signs: yes. So (10, 10) is feasible, with $z = 40 \times 10 + 60 \times 10 = 1000$ thousand DA (one million DA).

Example 2. Is $(15, 15)$ feasible? Assembly: $15 + 30 = 45 > 40$. No. One violated constraint is enough: we do not need to check the others.

Testing some plans.

Plan (x_1, x_2)	Assembly (≤ 40)	Testing (≤ 60)	Cameras (≤ 18)	Feasible?	z (thousand DA)
$(0, 0)$	0	0	0	yes	0
$(20, 0)$	20	60	0	yes	800
$(0, 18)$	36	36	18	yes	1080
$(10, 10)$	30	50	10	yes	1000
$(15, 15)$	45	75	15	no	—
$(4, 18)$	40	48	18	yes	1240

The best plan in this table gives 1240 thousand DA. But the variables can take infinitely many values, so we cannot test all the plans: we need a method. This is the subject of Lecture 2.

Check yourself

Q1. Is the plan $(12, 12)$ feasible? If yes, what is its profit?

Q2. Is the plan $(20, 2)$ feasible?

A plan is a point of the plane. With two variables, each plan (x_1, x_2) is a point. Each constraint is a line, and the feasible plans form a region (Figure 1). The plan $(15, 15)$ is outside the region, because it violates the assembly constraint. The best plans of the table are on the border of the region. Finding the best plan means finding the best point of the region: this is what the graphical method of Lecture 2 does.

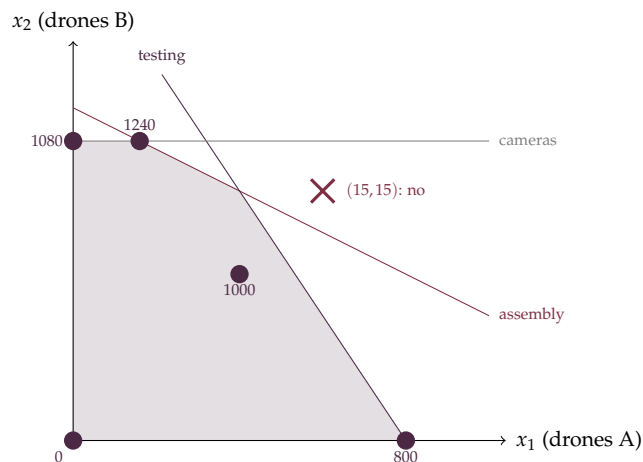


Figure 1: The plans of the table as points, with their profit. The shaded region contains the feasible plans; the cross $(15, 15)$ is not feasible.

2.7 Whole numbers: lucky or not?

In Lecture 2 we will find that the best plan of the LP relaxation is $(x_1, x_2) = (10, 15)$, with $z = 1300$ thousand DA. These are whole numbers, so this plan is also the best plan of the real problem: no whole-number plan can beat the best real-number plan. Here we are *lucky*: this is a coincidence, and it is *not guaranteed* for other production problems.

Example. Suppose the workshop has 61 testing hours instead of 60. Then:

	Plan (x_1, x_2)	z (thousand DA)
Best plan of the LP relaxation	$(10.5, 14.75)$: not whole numbers	1305
Rounding up	$(11, 15)$: assembly $11 + 30 = 41 > 40$, not feasible	—
Rounding down	$(10, 14)$	1240
Best whole-number plan	$(10, 15)$	1300

So we cannot simply round the LP answer: rounding can give a plan that is not feasible, or a plan that is not the best. When whole numbers are necessary, we check the LP answer, and if it is not made of whole numbers we need *integer programming* (another subject). The LP relaxation is still very useful: its optimal value (1305) is an upper bound for the real problem, because every whole-number plan is also a real-number plan.

Check yourself

Q3. With 61 testing hours, check that $(10.5, 14.75)$ satisfies the three constraints, and that its profit is 1305 thousand DA.

3 What does “linear” mean?

Allowed (linear)	Not allowed (not linear)
a number times a variable: $40x_1, -2x_3, 0.5x_2$	a product of variables: x_1x_2
sums of such terms: $40x_1 + 60x_2$	powers and roots: $x_1^2, \sqrt{x_1}$
a constant added: $3x_1 + 2x_2 + 10$	division by a variable: $1/x_1$
	absolute values: $ x_1 - x_2 $

Examples. $3x_1 - x_2 \leq 5$ is linear. $x_1x_2 \geq 4$ is not linear (product). $x_1 + 2 = 7x_2$ is linear: it is the same as $x_1 - 7x_2 = -2$. Section 10 shows tricks to transform some non-linear conditions into linear ones.

A word about constants. Strictly speaking, an expression with a constant, like $3x_1 + 2x_2 + 10$, is called *affine*. Constants are no problem in a linear program: in an objective, a constant does not change the best plan (it only adds 10 to the value); in a constraint, we move it to the right side.

Comparison signs

A linear program uses only the signs $\leq$, $=$ and $\geq$. Strict inequalities $<$ and $>$ are *not allowed*, even if the expression is linear: $x_1 < 5$ is a linear inequality, but it cannot appear in an LP.

Why? With a strict inequality the best plan may not exist. Example: maximise x_1 subject to $x_1 < 5$. The plan 4.9 is feasible, 4.99 is better, 4.999 is better again, and there is no best plan (5 itself is not allowed). Solvers also accept only $\leq$, $=$ and $\geq$: for example, the Gurobi solver does not support strict inequalities. In practice we write $x_1 \leq 5$, or $x_1 \leq 4.99$ if the value 5 is really forbidden.

4 A method in five steps

Method 1.1 — Write a linear program

1. **Read** the text twice. Put all the numbers in a table, with units.
2. **Variables:** what do we decide? Write the meaning of each variable, with unit and period.
3. **Objective:** what do we want? Write z , and say max or min.

4. **Constraints:** what limits us? One constraint for each limit, each requirement and each balance.
5. **Check:** sign constraints, units in each constraint, every number of the text used, everything linear.

From words to constraints.

In the text	Constraint
"at most b ", "not more than b ", "we have b "	$\dots \leq b$
"at least b ", "not less than b ", "we need b "	$\dots \geq b$
"exactly b ", "all of it must be used"	$\dots = b$
" x is at least twice y "	$x \geq 2y$, i.e. $x - 2y \geq 0$
"at least 30% of the total is x_1 "	$x_1 \geq 0.3(x_1 + x_2 + x_3)$ (Rule 1.1)
" x and y differ by at most 5"	$x - y \leq 5$ and $y - x \leq 5$ (Rule 1.7)
"what comes in = what goes out"	a balance: $\dots = \dots$

Good habit: put all the variables on the left side and the numbers on the right side.

5 Example 1.2: mixing raw materials

The problem. A factory makes exactly 1000 kg of a special material for robot arms, by mixing three raw materials, called M1, M2 and M3. Each raw material contains two important ingredients, called A and B. The final material must contain between 18% and 20% of ingredient A, and at least 8% of ingredient B. Find the cheapest mixture.

Raw material	Ingredient A	Ingredient B	Cost (DA/kg)	Available
M1	10%	2%	80	no limit
M2	5%	30%	600	300 kg
M3	60%	0%	250	no limit

Step 2: variables. x_1, x_2, x_3 = kilograms of M1, M2 and M3 in the mixture.

Step 3: objective. The cost is $80x_1 + 600x_2 + 250x_3$ dinars, to minimise.

Step 4: constraints.

- ▶ Total mass: exactly 1000 kg: $x_1 + x_2 + x_3 = 1000$.
- ▶ Ingredient A. 10% of x_1 kilograms is $0.10x_1$ kilograms of ingredient A; in the same way $0.05x_2$ and $0.60x_3$. So the mixture contains $0.10x_1 + 0.05x_2 + 0.60x_3$ kilograms of ingredient A. It must be between 18% and 20% of 1000 kg, that is between 180 and 200 kg. "Between" gives two constraints:

$$0.10x_1 + 0.05x_2 + 0.60x_3 \geq 180, \quad 0.10x_1 + 0.05x_2 + 0.60x_3 \leq 200.$$

- ▶ Ingredient B: at least 8% of 1000 kg, that is 80 kg: $0.02x_1 + 0.30x_2 \geq 80$.
- ▶ Available M2: $x_2 \leq 300$. Signs: $x_1, x_2, x_3 \geq 0$.

The model.

$$\begin{aligned} \min \quad & z = 80x_1 + 600x_2 + 250x_3 \\ \text{s.t.} \quad & x_1 + x_2 + x_3 = 1000 \\ & 0.10x_1 + 0.05x_2 + 0.60x_3 \geq 180 \\ & 0.10x_1 + 0.05x_2 + 0.60x_3 \leq 200 \\ & 0.02x_1 + 0.30x_2 \geq 80 \\ & x_2 \leq 300, \quad x_1, x_2, x_3 \geq 0 \end{aligned}$$

Here the masses can really be any real numbers, so this LP is the real problem (it is not a relaxation).

When the total is not fixed. Suppose now that the factory also decides how much material to make. Then “at least 18% of ingredient A” gives a quotient, which is not linear:

$$\frac{0.10x_1 + 0.05x_2 + 0.60x_3}{x_1 + x_2 + x_3} \geq 0.18.$$

Rule 1.1 — Percentages of a total

If the total is positive, multiply both sides by the total (an inequality does not change direction when we multiply by a positive number), then bring all terms to the left.

Here: $0.10x_1 + 0.05x_2 + 0.60x_3 \geq 0.18x_1 + 0.18x_2 + 0.18x_3$, and since $0.10 - 0.18 = -0.08$, $0.05 - 0.18 = -0.13$ and $0.60 - 0.18 = 0.42$:

$$-0.08x_1 - 0.13x_2 + 0.42x_3 \geq 0 \quad (\text{linear}).$$

Check with numbers. Take $x_1 = 500$, $x_2 = 200$, $x_3 = 300$ (total 1000 kg). Ingredient A: $50 + 10 + 180 = 240$ kg, that is 24%, at least 18%. The linear form gives $-40 - 26 + 126 = 60 \geq 0$. Both forms agree.

6 Example 1.3: transport

The problem. Two warehouses W1 and W2 store spare batteries: 300 in W1 and 400 in W2. Three sites S1, S2, S3, where robots work, need 200, 250 and 200 batteries. The cost of sending one battery is:

Cost (DA per battery)	S1	S2	S3	Stock
W1	800	600	1000	300
W2	900	1200	700	400
Need	200	250	200	

Find the cheapest transport plan.

Variables. We decide how many batteries travel on each route. There are $2 \times 3 = 6$ routes: x_{ij} = number of batteries sent from warehouse W_i to site S_j ($i = 1, 2$ and $j = 1, 2, 3$). For example, x_{12} is the number sent from W1 to S2.

The model.

$$\begin{aligned} \min \quad & z = 800x_{11} + 600x_{12} + 1000x_{13} + 900x_{21} + 1200x_{22} + 700x_{23} \\ \text{s.t.} \quad & x_{11} + x_{12} + x_{13} \leq 300 && (\text{what leaves W1}) \\ & x_{21} + x_{22} + x_{23} \leq 400 && (\text{what leaves W2}) \\ & x_{11} + x_{21} \geq 200 && (\text{what S1 receives}) \\ & x_{12} + x_{22} \geq 250 && (\text{what S2 receives}) \\ & x_{13} + x_{23} \geq 200 && (\text{what S3 receives}) \\ & x_{ij} \geq 0 \text{ for all } i, j \end{aligned}$$

The sum symbol. $\sum_{j=1}^3 x_{1j}$ means $x_{11} + x_{12} + x_{13}$. With this symbol the model can be written for any number of warehouses and sites: $\min \sum_i \sum_j c_{ij} x_{ij}$ subject to $\sum_j x_{ij} \leq s_i$ (stock of warehouse i), $\sum_i x_{ij} \geq d_j$ (need of site j) and $x_{ij} \geq 0$.

A simple check. The total stock is $300 + 400 = 700$ and the total need is $200 + 250 + 200 = 650$. Since $650 \leq 700$, there are enough batteries, and a feasible plan exists. For example: W1 sends 200 to S1 and 100 to S2, and W2 sends 150 to S2 and 200 to S3. W1 sends 300 (its whole stock), W2 sends 350, and each site receives exactly its need. The cost is $200 \times 800 + 100 \times 600 + 150 \times 1200 + 200 \times 700 = 540\,000$ DA.

If the sites needed more than 700 batteries in total, no plan could satisfy all the needs: the problem would have no feasible solution (we say it is *infeasible*).

Check yourself

Q4. Give another feasible plan for Example 1.3 and compute its cost. Is it cheaper than 540 000 DA?

7 Example 1.4: assignment

The problem. Three robots R1, R2, R3 must do three tasks T1, T2, T3. Each robot does exactly one task, and each task is done by exactly one robot. The times (in minutes) are:

Time (minutes)	T1	T2	T3
R1	9	2	7
R2	6	4	3
R3	5	8	1

Find the assignment with the smallest total time.

How many assignments? For T1 we can choose 3 robots; then 2 robots remain for T2; then 1 for T3. So there are $3 \times 2 \times 1 = 6$ assignments. With n robots and n tasks there are $n! = n \times (n-1) \times \dots \times 1$ assignments: $10! = 3\,628\,800$ and $20! \approx 2.4 \times 10^{18}$. We cannot try them all.

Variables. $x_{ij} = 1$ if robot R_i does task T_j , and $x_{ij} = 0$ otherwise (9 variables).

The model.

$$\min z = 9x_{11} + 2x_{12} + 7x_{13} + 6x_{21} + 4x_{22} + 3x_{23} + 5x_{31} + 8x_{32} + 1x_{33}$$

subject to

- ▶ each robot does one task: $x_{11} + x_{12} + x_{13} = 1$, $x_{21} + x_{22} + x_{23} = 1$, $x_{31} + x_{32} + x_{33} = 1$;
- ▶ each task is done by one robot: $x_{11} + x_{21} + x_{31} = 1$, $x_{12} + x_{22} + x_{32} = 1$, $x_{13} + x_{23} + x_{33} = 1$;
- ▶ $x_{ij} \geq 0$.

Only the chosen pairs (with $x_{ij} = 1$) count in z . The condition " $x_{ij} = 0$ or 1 " is not linear, so we write only $x_{ij} \geq 0$; the equalities then give $x_{ij} \leq 1$. We accept the following fact without proof: *for this problem, the linear program always has an optimal solution with all values equal to 0 or 1*, so it gives a real assignment.

8 Writing a linear program with matrices

8.1 Why do we need matrices?

The drone workshop has 2 variables and 3 constraints, so we can write everything in full. The transport example (Example 1.3) already has 6 variables and 5 constraints. Real problems are much larger: one variable for each route, each product or each hour of the day gives thousands of variables. We need a *short way to write any linear program*, whatever its size. Matrices give this short way.

Matrices also show the *structure* of a problem: we will see that the data table of a problem is exactly a matrix. Finally, computers store the data of a linear program as matrices, and the methods of the next lectures are written with them.

We only need three things from linear algebra: columns of numbers, the product of a line by a column, and the product of a matrix by a column. We recall them now on the drone workshop.

8.2 Columns, lines, and the product of a line by a column

A list of numbers written vertically is a *column* (or *vector*). For the drone workshop we use two columns: the plan x and the profits c :

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad c = \begin{pmatrix} 40 \\ 60 \end{pmatrix}.$$

The *transpose* $c^\top$ is the same list written horizontally, as a line: $c^\top = (40 \ 60)$.

To multiply a line by a column of the same length, we multiply the numbers that are in the same position, and we add the results:

$$c^\top x = (40 \ 60) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 40x_1 + 60x_2.$$

This is exactly our objective function: *the profit is* $z = c^\top x$. For example, for the plan $x = (10, 5)$ we find $c^\top x = 40 \times 10 + 60 \times 5 = 700$, as in Section 2.3.

In general, with n variables, $c^\top x = c_1x_1 + c_2x_2 + \dots + c_nx_n = \sum_{j=1}^n c_jx_j$. The number c_j is the profit (or the cost) of one unit of variable j .

8.3 A matrix times a column

Now we put the coefficients of the constraints in a table, with *one line for each constraint* and *one column for each variable*. This table is the matrix A . We put the right-hand sides in a column b :

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 2 \\ 0 & 1 \end{pmatrix} \begin{matrix} \leftarrow \text{assembly} \\ \leftarrow \text{testing} \\ \leftarrow \text{cameras} \end{matrix} \quad b = \begin{pmatrix} 40 \\ 60 \\ 18 \end{pmatrix}.$$

To compute the product Ax , we multiply *each line* of A by the column x , as in the previous subsection:

$$Ax = \begin{pmatrix} 1 & 2 \\ 3 & 2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \cdot x_1 + 2 \cdot x_2 \\ 3 \cdot x_1 + 2 \cdot x_2 \\ 0 \cdot x_1 + 1 \cdot x_2 \end{pmatrix} = \begin{pmatrix} x_1 + 2x_2 \\ 3x_1 + 2x_2 \\ x_2 \end{pmatrix}.$$

The three lines of Ax are exactly the left sides of our three constraints.

Example with numbers. For the plan $x = (10, 10)$:

$$Ax = \begin{pmatrix} 1 \times 10 + 2 \times 10 \\ 3 \times 10 + 2 \times 10 \\ 0 \times 10 + 1 \times 10 \end{pmatrix} = \begin{pmatrix} 30 \\ 50 \\ 10 \end{pmatrix}.$$

These are the assembly hours, the testing hours and the cameras used by this plan. This is true in general: Ax is what the plan x uses of each resource.

Sizes. If A has m lines and n columns (we say that A is an $m \times n$ matrix), then x must have n lines (one per variable), and Ax has m lines (one per constraint). Here A is a 3×2 matrix, x has 2 lines and Ax has 3 lines.

8.4 Comparing two columns

For two columns u and v of the same size, we write $u \leq v$ when *each line* of u is at most the same line of v : $u_1 \leq v_1$, $u_2 \leq v_2$, and so on. In the same way for $\geq$ and $=$. So $u \leq v$ is a short way to write a list of inequalities.

For example, $(30, 50, 10) \leq (40, 60, 18)$ is true, because $30 \leq 40$, $50 \leq 60$ and $10 \leq 18$. So the plan $(10, 10)$ satisfies the three constraints: this is the computation of Example 1 in Section 2.6, written in one line.

Therefore $Ax \leq b$ says that *all* the constraints hold, and $x \geq 0$ (where 0 is the column of zeros) says that $x_1 \geq 0$ and $x_2 \geq 0$.

Remark. Two columns cannot always be compared: neither $(2, 5) \leq (3, 4)$ nor $(3, 4) \leq (2, 5)$ is true. This is not a problem: we only use the sign $\leq$ between columns to write lists of constraints.

8.5 The drone workshop in matrix form

With these notations, the drone workshop is

$$\max c^\top x \quad \text{subject to} \quad Ax \leq b, \quad x \geq 0.$$

The lines and the columns of A have a clear meaning (Figure 2):

- ▶ **Line i** of A , with the number b_i , is *constraint i* : what each product uses of resource i , and how much of this resource we have.
- ▶ **Column j** of A , with the number c_j , describes *variable j* : what one unit of product j uses of each resource, and its profit. For example, column 1 says that one drone A uses 1 hour of assembly, 3 hours of testing and no camera, and brings 40 thousand DA.
- ▶ The number a_{ij} in line i and column j is the quantity of resource i used by one unit of product j . For example, $a_{21} = 3$: one drone A needs 3 hours of testing.

	drone A	drone B	available
Assembly (h)	1	2	40
Testing (h)	3	2	60
Cameras	0	1	18
Profit	40	60	

matrix A : one line per resource,
one column per product

column b : what we have

line $c^\top$: the profits

Figure 2: The data table of the drone workshop (Section 2.1) is exactly the matrix A , the column b and the line $c^\top$. Writing the matrices is copying the table.

8.6 Any linear program, with sums and with matrices

Consider now a linear program with n variables $x_1, \dots, x_n$ and m constraints of type “ $\leq$ ”. We call a_{ij} the coefficient of variable j in constraint i . With the sum symbol of Section 6, the program is

$$\max \sum_{j=1}^n c_j x_j \quad \text{subject to} \quad \sum_{j=1}^n a_{ij} x_j \leq b_i \quad (i = 1, \dots, m), \quad x_j \geq 0 \quad (j = 1, \dots, n).$$

With A the $m \times n$ matrix of the numbers a_{ij} , line i of Ax is exactly $\sum_j a_{ij} x_j$. So the same program is simply

$$\max c^\top x \quad \text{subject to} \quad Ax \leq b, \quad x \geq 0.$$

This short formula is valid for 2 variables or for 10 000 variables: only the sizes of c , A , b and x change.

8.7 Constraints with different signs

Many problems mix the signs $\leq$, $\geq$ and $=$. Then we group the constraints by sign. For Example 1.2 (mixing), with $x = (x_1, x_2, x_3)$:

$$\min c^\top x \quad \text{subject to} \quad A_1 x = b_1, \quad A_2 x \geq b_2, \quad A_3 x \leq b_3, \quad x \geq 0,$$

where

$$c = \begin{pmatrix} 80 \\ 600 \\ 250 \end{pmatrix}, \quad A_1 = (1 \ 1 \ 1), \quad b_1 = 1000 \quad (\text{total mass}),$$

$$A_2 = \begin{pmatrix} 0.10 & 0.05 & 0.60 \\ 0.02 & 0.30 & 0 \end{pmatrix}, \quad b_2 = \begin{pmatrix} 180 \\ 80 \end{pmatrix} \quad \begin{array}{l} \leftarrow \text{A at least} \\ \leftarrow \text{B at least} \end{array}$$

$$A_3 = \begin{pmatrix} 0.10 & 0.05 & 0.60 \\ 0 & 1 & 0 \end{pmatrix}, \quad b_3 = \begin{pmatrix} 200 \\ 300 \end{pmatrix} \quad \begin{array}{l} \leftarrow \text{A at most} \\ \leftarrow \text{M2 available} \end{array}$$

This is correct, but heavy: three matrices and three kinds of constraints. It would be simpler to have *one single kind* of constraint. This is the reason for the next section: we learn to write every linear program in one fixed form.

Check yourself

Q5. For the drone workshop, compute Ax for the plan $x = (4, 18)$. Is the plan feasible? Which resources are fully used?

Q6. Write the transport problem of Example 1.3 in matrix form, with the variables in the order $x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}$. (Group the “ $\leq$ ” constraints and the “ $\geq$ ” constraints.)

9 Different forms of the same linear program

9.1 Why several forms?

The same problem can be written in many ways: we can maximise or minimise; the constraints can be “ $\leq$ ”, “ $\geq$ ” or “ $=$ ”; the variables can be non-negative or have any sign. A method of solution, like the simplex method of Lecture 3, is much easier to describe for *one fixed form*. So we choose two standard ways of writing a linear program, and we learn rules to transform *any* linear program into them.

The important point is this: *the rules change the way we write the problem, not the problem itself*. Every feasible plan of the old problem gives a feasible plan of the new problem, with the same value of the objective (up to the sign, see Rule 1.2), and conversely. So after solving the new problem, we can always come back to the answer of the old one.

Definition 1.3 — Canonical form

$\max c^T x$ subject to $Ax \leq b$, $x \geq 0$. All constraints are “ $\leq$ ”, all variables are ≥ 0 , and we maximise.

Definition 1.4 — Standard form

$\max c^T x$ subject to $Ax = b$, $x \geq 0$. All constraints are equalities, all variables are ≥ 0 , and we maximise.

The canonical form is natural for problems with limited resources (“we use at most what we have”); the drone workshop is already in canonical form. The standard form is the form used by the simplex method, because equalities can be handled with the tools of linear algebra (systems of linear equations).

9.2 Five rules to change the form

For each rule we give the rule, the reason why it does not change the problem, and an example.

Rule 1.2 — Minimum into maximum

$\min z$ is the same as $\max (-z)$: the best plan is the same, and the optimal value only changes sign.

Why? Suppose three plans give $z = 5$, $z = 7$ and $z = 9$. Then $-z$ takes the values -5 , -7 and -9 . The plan with the *smallest* z (here 5) is exactly the plan with the *largest* $-z$ (here -5). This is true for any list of numbers, so both problems have the same best plans, and $\min z = -\max(-z)$.

Example. $\min 3x_1 + x_2$ becomes $\max -3x_1 - x_2$. If the minimum of $3x_1 + x_2$ is 5, then the maximum of $-3x_1 - x_2$ is -5 , at the same plan.

Do not forget: at the end, change the sign of the optimal value again to get the answer of the original problem.

Rule 1.3 — Changing the direction of an inequality

Multiply both sides by -1 : “ $\geq$ ” becomes “ $\leq$ ”.

Why? Multiplying both sides of an inequality by a *negative* number reverses the inequality: $7 \geq 2$, and $-7 \leq -2$. So a plan satisfies $x_1 + x_2 \geq 2$ exactly when it satisfies $-x_1 - x_2 \leq -2$: the two constraints accept the same plans.

Example. $x_1 + x_2 \geq 2$ becomes $-x_1 - x_2 \leq -2$. Check with the plan $(1, 3)$: $1 + 3 = 4 \geq 2$, and $-1 - 3 = -4 \leq -2$. Both say “yes”.

Be careful: multiply *all* the terms, on both sides, including the right-hand side.

Rule 1.4 — An equality is two inequalities

$a = b$ is the same as “ $a \leq b$ and $a \geq b$ ”.

Why? A number which is at most b and at least b is equal to b .

Example. $x_1 - x_2 = 1$ becomes $x_1 - x_2 \leq 1$ and $x_1 - x_2 \geq 1$. For the canonical form we then apply Rule 1.3 to the second one: $-x_1 + x_2 \leq -1$.

Rule 1.5 — A free variable

A variable that may be negative is called *free*. Write it as the difference of two non-negative variables: $x_j = x_j^+ - x_j^-$ with $x_j^+ \geq 0$, $x_j^- \geq 0$, and replace x_j everywhere.

Why? Every real number is the difference of two non-negative numbers: $3 = 3 - 0$ and $-2 = 0 - 2$. In general, if $x_j \geq 0$ take $x_j^+ = x_j$ and $x_j^- = 0$; if $x_j < 0$ take $x_j^+ = 0$ and $x_j^- = -x_j$. Conversely, a difference $x_j^+ - x_j^-$ can be any real number. So the new variables describe exactly the same plans. The pair is not unique ($3 = 3 - 0 = 5 - 2$), but this is no problem, because only the difference appears in the problem.

When is a variable free? For example, a change of stock (it can go up or down), a temperature, or a position on an axis.

A variable ≤ 0 . If a variable can only be negative or zero, write $x_j = -x'_j$ with $x'_j \geq 0$.

Rule 1.6 — Slack and surplus variables

A constraint " $\leq$ " becomes an equality if we *add* a slack variable $s \geq 0$. A constraint " $\geq$ " becomes an equality if we *subtract* a surplus variable $e \geq 0$.

Why? For a plan (x_1, x_2) , define $s_1 = 40 - (x_1 + 2x_2)$. Then

$$x_1 + 2x_2 \leq 40 \iff 40 - (x_1 + 2x_2) \geq 0 \iff s_1 \geq 0,$$

and $x_1 + 2x_2 + s_1 = 40$ by the definition of s_1 . So " $x_1 + 2x_2 \leq 40$ " is the same as " $x_1 + 2x_2 + s_1 = 40$ with $s_1 \geq 0$ ". In the same way, for $x_1 + x_2 \geq 2$ we define $e_1 = (x_1 + x_2) - 2$, and the constraint is the same as " $x_1 + x_2 - e_1 = 2$ with $e_1 \geq 0$ ".

Meaning. The slack s_1 is the *unused* quantity (here, unused assembly hours). The surplus e_1 is the quantity *above* the requirement.

Be careful: each constraint has its own slack or surplus variable, and these variables have the coefficient 0 in the objective (unused hours bring no profit).

9.3 The rules in practice

To write a linear program in canonical form or in standard form, we always work in the same order:

1. **Objective.** If it is a minimisation, use Rule 1.2.
2. **Variables.** Replace each free variable using Rule 1.5 (and each variable ≤ 0 by $-x'_j$).
3. **Constraints.**
 - ▶ For the *canonical form*: replace each " $=$ " by two inequalities (Rule 1.4), then turn each " $\geq$ " into " $\leq$ " (Rule 1.3).
 - ▶ For the *standard form*: add a slack variable to each " $\leq$ ", subtract a surplus variable from each " $\geq$ " (Rule 1.6). Equalities stay as they are.
4. **Check.** Every variable is ≥ 0 ; every constraint has the right sign; the objective is a maximisation.
5. **After solving, come back** to the original problem: $x_j = x_j^+ - x_j^-$, and change the sign of the optimal value if Rule 1.2 was used.

9.4 Worked example (a): the drone workshop in standard form

We add one slack variable to each constraint: s_1 = unused assembly hours, s_2 = unused testing hours, s_3 = unused cameras.

$$\begin{aligned} \max \quad & 40x_1 + 60x_2 \\ \text{s.t.} \quad & x_1 + 2x_2 + s_1 = 40, \quad 3x_1 + 2x_2 + s_2 = 60, \quad x_2 + s_3 = 18, \\ & x_1, x_2, s_1, s_2, s_3 \geq 0. \end{aligned}$$

For the plan $(10, 10)$, the slacks are $s = b - Ax = (40, 60, 18) - (30, 50, 10) = (10, 10, 8)$: 10 unused assembly hours, 10 unused testing hours and 8 unused cameras. For the plan $(15, 15)$ we find $s_1 = 40 - 45 = -5 < 0$: a negative slack shows a violated constraint, so this plan is not feasible.

In matrix form, the new matrix is the old matrix A followed by the identity matrix I (one column per slack variable):

$$\begin{pmatrix} 1 & 2 & 1 & 0 & 0 \\ 3 & 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} = \begin{pmatrix} 40 \\ 60 \\ 18 \end{pmatrix}.$$

The identity matrix in the columns of the slack variables will be very useful in Lecture 3.

9.5 Worked example (b): from a general linear program to the canonical form

$$\min 3x_1 + 2x_2 \quad \text{s.t.} \quad x_1 + x_2 \geq 2, \quad x_1 - 2x_2 \leq 4, \quad x_1 \geq 0, \quad x_2 \text{ free.}$$

We follow the order of Section 9.3.

1. *Objective* (Rule 1.2): $\min 3x_1 + 2x_2$ becomes $\max -3x_1 - 2x_2$.
2. *Variables* (Rule 1.5): $x_2 = x_2^+ - x_2^-$. We replace x_2 everywhere:
 - ▶ objective: $-3x_1 - 2(x_2^+ - x_2^-) = -3x_1 - 2x_2^+ + 2x_2^-$;
 - ▶ first constraint: $x_1 + x_2^+ - x_2^- \geq 2$;
 - ▶ second constraint: $x_1 - 2(x_2^+ - x_2^-) \leq 4$, that is $x_1 - 2x_2^+ + 2x_2^- \leq 4$.
3. *Constraints* (Rule 1.3 on the first one): $-x_1 - x_2^+ + x_2^- \leq -2$. The second one is already " $\leq$ ".

Canonical form:

$$\begin{aligned} \max \quad & -3x_1 - 2x_2^+ + 2x_2^- \\ \text{s.t.} \quad & -x_1 - x_2^+ + x_2^- \leq -2, \quad x_1 - 2x_2^+ + 2x_2^- \leq 4, \quad x_1, x_2^+, x_2^- \geq 0. \end{aligned}$$

(The standard form of this example is Exercise 7 of TD 1.)

The best plan. This problem has a best plan: $(x_1, x_2) = (0, 2)$, with $z = 4$. Here is a short proof. The first constraint gives $x_2 \geq 2 - x_1$, so for every feasible plan

$$3x_1 + 2x_2 \geq 3x_1 + 2(2 - x_1) = x_1 + 4 \geq 4 \quad (\text{because } x_1 \geq 0),$$

and the plan $(0, 2)$ is feasible ($0 + 2 \geq 2$ and $0 - 4 \leq 4$) with value exactly 4.

Coming back from the canonical form. In the canonical form, the plan $x_1 = 0, x_2^+ = 2, x_2^- = 0$ gives the value -4 , and so do $x_2^+ = 3, x_2^- = 1$, or any pair with $x_2^+ - x_2^- = 2$. The canonical problem has many best plans, but they all give the same original plan $x_2 = 2$. Its optimal value is -4 , so the minimum of the original problem is $-(-4) = 4$ (Rule 1.2).

Warning: changing the form does not tell whether a best plan exists

Keep the same constraints, but take the objective $\min 3x_1 - x_2$. For $x_1 = 0$ and $x_2 = t$ with $t \geq 2$, both constraints hold ($0 + t \geq 2$ and $0 - 2t \leq 4$), and the objective is $-t$. When t grows, $-t$ goes to $-\infty$: there is no smallest value. This linear program has no best plan; we say that it is *unbounded* (Lecture 2, Case 4). The rules transform it in the same way, but the new problem has no best plan either. Changing the form never creates or removes a best plan.

Check yourself

Q7. Write the mixing problem of Example 1.2 in canonical form.

Q8. Consider $\min 2x_1 + x_2$ s.t. $x_1 + x_2 \geq 1$, $x_1 - x_2 = 3$, $x_1 \geq 0$, x_2 free.

(a) Write it in standard form. (b) Check that the plan $(2, -1)$ is feasible, and give the values of all the variables of the standard form for this plan.

10 Useful tricks, and what is impossible

Some conditions do not look linear, but they can be written with linear constraints. Others can never be written with linear constraints. This section shows both, and gives a simple test to recognise the impossible ones.

10.1 Absolute values

Rule 1.7 — Absolute value “at most”

$|a| \leq b$ is the same as “ $a \leq b$ and $-a \leq b$ ”. For example, $|x_1 - x_2| \leq 5$ becomes $x_1 - x_2 \leq 5$ and $x_2 - x_1 \leq 5$.

Why? The absolute value of a number is the larger of the number and its opposite: $|6| = \max(6, -6) = 6$ and $|-6| = \max(-6, 6) = 6$. The larger of two numbers is at most 5 exactly when *both* numbers are at most 5.

Check. $x = (8, 2)$: $|8 - 2| = 6 > 5$, and indeed $x_1 - x_2 = 6 > 5$: the linear form also says “no”. $x = (4, 7)$: $|4 - 7| = 3 \leq 5$, and $4 - 7 = -3 \leq 5$, $7 - 4 = 3 \leq 5$: both forms say “yes”.

Example of use. “The two robots must do almost the same number of tasks: the difference is at most 5” gives $|x_1 - x_2| \leq 5$, so two linear constraints.

10.2 The smallest or the largest of several values

Rule 1.8 — Maximise the smallest value

To maximise $\min(x_1, x_2)$, add a new variable t : maximise t subject to $t \leq x_1$, $t \leq x_2$ (and the other constraints).

Why? The constraints say that t is at most x_1 and at most x_2 , so t is at most the smaller of the two. Since we maximise t , the best plan pushes t up until it touches the smaller one. For example, if $x_1 = 3$ and $x_2 = 7$, the constraints give $t \leq 3$ and $t \leq 7$, and the largest possible t is $3 = \min(3, 7)$.

Example of use. Two robots share the work, and we want to be fair: we maximise the work of the robot that works less.

The opposite case. To minimise the largest value $\max(x_1, x_2)$, minimise t subject to $t \geq x_1$ and $t \geq x_2$.

Minimising an absolute value. Since $|a| = \max(a, -a)$, the same idea works: to minimise $|x_1 - x_2|$, minimise t subject to $t \geq x_1 - x_2$ and $t \geq x_2 - x_1$.

10.3 What is impossible, and how to recognise it

Some conditions can *never* be written with linear constraints. To recognise them we use a simple property.

Key fact 1.1 — Linear constraints keep middle points

If two plans P and Q satisfy some linear constraints, then their middle point $M = \frac{P+Q}{2}$ also satisfies them.

Why? Take a linear constraint $a_1x_1 + a_2x_2 \leq b$. Its left side at M is

$$a_1 \frac{p_1 + q_1}{2} + a_2 \frac{p_2 + q_2}{2} = \frac{(a_1p_1 + a_2p_2) + (a_1q_1 + a_2q_2)}{2},$$

the average of its values at P and at Q . Both values are $\leq b$, so their average is also $\leq b$. The same computation works for “ $\geq$ ”, for “ $=$ ”, and for any number of variables. It also works for every point of the segment from P to Q , not only the middle: the value at such a point is a weighted average of the values at P and Q , so it is still $\leq b$.

The middle-point test. If you find two points that satisfy a condition, but their middle point does not, then this condition can *not* be written with linear constraints. Three examples:

- ▶ *A product:* $x_1x_2 \leq 1$ (with $x_1, x_2 \geq 0$). The points $P = (2, 0.5)$ and $Q = (0.5, 2)$ satisfy it ($x_1x_2 = 1$), but their middle point $M = (1.25, 1.25)$ gives $1.25 \times 1.25 = 1.5625 > 1$.
- ▶ *An absolute value “at least”:* $|x_1 - x_2| \geq 5$. The points $(5, 0)$ and $(0, 5)$ satisfy it, but their middle point $(2.5, 2.5)$ gives $|0| = 0 < 5$. This is why Rule 1.7 only works for “at most”.
- ▶ *A condition with “or”:* “ $x_1 = 0$ or $x_1 \geq 10$ ” (a machine is either off, or it produces at least 10 units). The values 0 and 10 satisfy it, but the middle value 5 does not.

Such conditions need other tools, mainly integer programming, which is not part of this lecture.

Check yourself

Q9. Write with linear constraints: “the difference between the hours of robot 1 (h_1) and robot 2 (h_2) is at most 3 hours”. Can you do the same for “at least 3 hours”? Use the middle-point test.

11 What to do with a strict inequality

11.1 The problem

In Section 3 we saw that a linear program uses only the signs $\leq$, $=$ and $\geq$. A strict inequality such as $x_1 < 5$ is linear, but it is not allowed, because with a strict inequality *the best plan may not exist*: to maximise x_1 subject to $x_1 < 5$, the plan 4.9 is good, 4.99 is better, 4.999 is better again, and no plan is the best. Solvers do not accept strict inequalities either.

But a text often says “less than”, “more than”, “strictly fewer than”. What should we do then? The answer depends on the kind of quantities.

11.2 Case 1: the quantities are whole numbers

Suppose the left side of the constraint is always a whole number: the variables count objects (drones, robots, tasks) and the coefficients are whole numbers. Then a strict inequality can be replaced *exactly*

by a non-strict one. For whole numbers, “ $k < 25$ ” means $k \leq 24$, and “ $k > 0$ ” means $k \geq 1$. So, if b is a whole number:

$$\text{expression} < b \iff \text{expression} \leq b - 1, \quad \text{expression} > b \iff \text{expression} \geq b + 1.$$

Examples (drone workshop, numbers of drones):

- ▶ “We make strictly fewer than 25 drones per week”: $x_1 + x_2 < 25$ becomes $x_1 + x_2 \leq 24$.
- ▶ “We make more drones A than drones B”: $x_1 > x_2$ becomes $x_1 \geq x_2 + 1$, that is $x_1 - x_2 \geq 1$.
- ▶ “We make at least some drones B”: $x_2 > 0$ becomes $x_2 \geq 1$.

This is not an approximation: for whole numbers, both constraints accept exactly the same plans. Then we solve the LP relaxation as usual (Section 2.5).

Effect on the drone workshop. The best plan of the workshop, (10, 15), makes exactly 25 drones. With “strictly fewer than 25 drones”, that is $x_1 + x_2 \leq 24$, it is no longer allowed. With the methods of Lecture 2 one finds the new best plan (8, 16): 24 drones and a profit of 1280 thousand DA instead of 1300.

11.3 Case 2: the quantities are real numbers

Now suppose that the quantities can take any real values: hours, kilograms, kilowatt-hours. We proceed in three steps.

Step 1: replace “ $<$ ” by “ $\leq$ ”, and solve. Very often the difference does not matter in practice: measures have a limited precision, and nobody can see the difference between 40 hours and 39.999 hours. Solvers themselves accept a constraint as satisfied when it is violated by a very small amount (often around 10^{-6}).

Step 2: look at the best plan. Two cases are possible.

- ▶ *The best plan satisfies the strict inequality.* Then it is also the best plan of the problem with the strict inequality, and we are done. *Why?* This plan is allowed in the strict problem. And no plan of the strict problem can be better, because every plan of the strict problem is also a plan of the problem with “ $\leq$ ”, where our plan is the best.
- ▶ *The best plans all satisfy the constraint with equality.* Then the problem with the strict inequality has *no best plan* (if at least one plan satisfies the strict inequality; if none does, the strict problem has no feasible plan at all): there are plans whose value is as close as we want to the best value, but no plan reaches it. *Why?* Let P be a best plan (with equality) and Q any plan with the strict inequality. Walk from Q towards P on the segment $[Q, P]$: the points of this segment, except P itself, satisfy the strict inequality and all the other constraints (Key fact 1.1), and their value increases regularly towards the best value (the values are weighted averages of the values at Q and P , and Q has a smaller value than P). So for every allowed plan there is a better allowed plan, closer to P , and no plan is the best.

Step 3: if there is no best plan, use a safety margin. We choose a small positive number ε that makes sense in the problem (the precision of a measure, the smallest amount that matters), and we write

$$\text{expression} \leq b - \varepsilon \quad \text{instead of} \quad \text{expression} < b.$$

This constraint is allowed in a linear program, and it really forbids the value b .

Method 1.2 — What to do with a strict inequality

1. **Whole numbers** (whole variables and whole coefficients): replace " $< b$ " by " $\leq b - 1$ " and " $> b$ " by " $\geq b + 1$ ". This is exact.
2. **Real numbers**: replace " $<$ " by " $\leq$ " (or " $>$ " by " $\geq$ ") and solve.
 - ▶ If the best plan satisfies the strict inequality, it is also the best plan of the strict problem.
 - ▶ If the best plan gives an equality, the strict problem has no best plan. If the value b is really forbidden, choose a margin $\varepsilon > 0$ from the meaning of the problem and write " $\leq b - \varepsilon$ " (or " $\geq b + \varepsilon$ ").

11.4 An example with real numbers

Suppose that the assembly team must work *strictly less than 40 hours* per week, and that drones can be counted with decimals (LP relaxation). The constraint is $x_1 + 2x_2 < 40$.

Step 1. With $x_1 + 2x_2 \leq 40$ we have the usual problem, whose best plan is $(10, 15)$ with $z = 1300$ (Lecture 2).

Step 2. At $(10, 15)$ the assembly time is $10 + 30 = 40$ hours: equality. This is the only best plan, so the strict problem has no best plan. Indeed, the plans $(10 - 10t, 15 - 15t)$ for a small $t > 0$ use $40 - 40t < 40$ assembly hours and give the profit $1300 - 1300t$. For $t = 0.01$ we get the plan $(9.9, 14.85)$: assembly $9.9 + 29.7 = 39.6$ hours, testing $29.7 + 29.7 = 59.4$ hours, profit $396 + 891 = 1287$. With $t = 0.001$ the profit is 1298.7, and so on: we get as close to 1300 as we want, but never 1300.

Step 3. If the team leader says that 39.5 hours is acceptable, we take $\varepsilon = 0.5$ hour and write $x_1 + 2x_2 \leq 39.5$. The best plan is then $(10.25, 14.625)$, with $z = 1287.5$ thousand DA (computed with the methods of Lecture 2).

An example where nothing changes. Suppose instead that the workshop must use *strictly fewer than 18 cameras*: $x_2 < 18$. The best plan $(10, 15)$ uses 15 cameras, and $15 < 18$: it satisfies the strict inequality, so it is still the best plan (Step 2, first case).

Check yourself

Q10. (a) The workshop must make strictly more drones A than drones B. Write this as a constraint of a linear program. (b) The workshop now has strictly fewer than 16 cameras (cameras are whole objects). Write the constraint. Is $(10, 15)$ still the best plan? Explain.

12 The four hypotheses of linear programming

A linear program is a simplified picture of reality. When we write one, we accept four hypotheses. It is important to know them, to see when a linear program is a good model and when it is not.

- ▶ **Proportionality.** Each term is proportional to its variable: two drones bring twice the profit of one drone and use twice the hours. *This is false* if there is a discount for large quantities, or a fixed cost to start a machine.
- ▶ **Additivity.** The total is the sum of the parts: the profit of the drones A does not depend on the number of drones B, and the hours used by A and by B add up. *This is false* if, for example, making the two models together needs extra cleaning time.
- ▶ **Divisibility.** The variables can take any real value: 10.5 drones is allowed by the model. *This*

is false for objects counted in whole numbers; then the linear program is only the LP relaxation (Section 2.5).

- ▶ **Certainty.** All the numbers of the model (profits, hours, quantities available) are known exactly. *This is never completely true:* prices change and times are estimated.

When a hypothesis is not true, we use other tools: *integer programming* when we need whole numbers, and *sensitivity analysis* (Chapter 3) to study what happens when the numbers change.

13 Summary

- ▶ A linear program has variables, a linear objective (max or min), linear constraints with $\leq$, $=$ or $\geq$, and sign constraints (Definition 1.1).
- ▶ Feasible solution, feasible set, optimal solution, optimal value (Definition 1.2).
- ▶ To write an LP, follow Method 1.1: read, variables, objective, constraints, check.
- ▶ Four classical models: production (drones), mixing (raw materials), transport (batteries), assignment (robots and tasks).
- ▶ Whole-number problems: the LP is a *relaxation*; its answer is whole only by luck (Section 2.7).
- ▶ Matrix form: $\max c^T x$ subject to $Ax \leq b$, $x \geq 0$. Line i of A is constraint i ; column j of A describes variable j ; Ax is what the plan x uses.
- ▶ Canonical form (Definition 1.3) and standard form (Definition 1.4); Rules 1.2 to 1.6 change the form without changing the problem.
- ▶ Tricks: percentages (Rule 1.1), absolute values “at most” (Rule 1.7), the smallest or largest value (Rule 1.8). Products, “at least” absolute values and “or” conditions are impossible (middle-point test, Key fact 1.1).
- ▶ Strict inequalities (Method 1.2): exact rewriting for whole numbers; for real numbers, solve with “ $\leq$ ”, check the best plan, and use a margin if needed.

What comes next. We can now write linear programs. In Lecture 2 we learn to *solve* them when there are two variables, by drawing, and we meet all the possible cases (one best plan, many best plans, no best plan).

Reading. F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*, Chapter 3; H. A. Taha, *Operations Research: An Introduction*, Chapter 2.

Answers to “Check yourself”

Q1. (12, 12): assembly $12 + 24 = 36 \leq 40$, testing $36 + 24 = 60 \leq 60$, cameras $12 \leq 18$: feasible. Profit $40 \times 12 + 60 \times 12 = 1200$ thousand DA.

Q2. (20, 2): assembly $20 + 4 = 24 \leq 40$, but testing $60 + 4 = 64 > 60$: not feasible.

Q3. Assembly $10.5 + 29.5 = 40 \leq 40$, testing $31.5 + 29.5 = 61 \leq 61$, cameras $14.75 \leq 18$. Profit $40 \times 10.5 + 60 \times 14.75 = 420 + 885 = 1305$ thousand DA.

Q4. Many answers. For example: W1 sends 50 to S1 and 250 to S2; W2 sends 150 to S1 and 200 to S3. Check: W1 sends 300, W2 sends 350; S1 receives $50 + 150 = 200$, S2 receives 250, S3 receives 200. Cost: $50 \times 800 + 250 \times 600 + 150 \times 900 + 200 \times 700 = 40\,000 + 150\,000 + 135\,000 + 140\,000 = 465\,000$ DA, cheaper than 540 000 DA. (In fact, this is the cheapest plan.)

Q5. $Ax = (4 + 36, 12 + 36, 18) = (40, 48, 18) \leq (40, 60, 18)$: feasible. Assembly (40 = 40) and cameras (18 = 18) are fully used; 12 testing hours are not used.

Q6. $\min c^\top x$ subject to $A_1x \leq b_1$, $A_2x \geq b_2$, $x \geq 0$, with $c = (800, 600, 1000, 900, 1200, 700)$ and

$$A_1 = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}, \quad b_1 = \begin{pmatrix} 300 \\ 400 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}, \quad b_2 = \begin{pmatrix} 200 \\ 250 \\ 200 \end{pmatrix}.$$

Each line of A_1 picks the routes that leave one warehouse (b_1 = the stocks); each line of A_2 picks the routes that arrive at one site (b_2 = the needs).

Q7. Rule 1.2 for the objective, Rule 1.4 for the equality, Rule 1.3 for each “ $\geq$ ”:

$$\begin{aligned} \max \quad & -80x_1 - 600x_2 - 250x_3 \\ \text{s.t.} \quad & x_1 + x_2 + x_3 \leq 1000, \quad -x_1 - x_2 - x_3 \leq -1000, \\ & -0.10x_1 - 0.05x_2 - 0.60x_3 \leq -180, \quad 0.10x_1 + 0.05x_2 + 0.60x_3 \leq 200, \\ & -0.02x_1 - 0.30x_2 \leq -80, \quad x_2 \leq 300, \quad x_1, x_2, x_3 \geq 0. \end{aligned}$$

Q8. (a) With $x_2 = x_2^+ - x_2^-$ and a surplus variable e_1 :

$$\max -2x_1 - x_2^+ + x_2^- \quad \text{s.t.} \quad x_1 + x_2^+ - x_2^- - e_1 = 1, \quad x_1 - x_2^+ + x_2^- = 3, \quad x_1, x_2^+, x_2^-, e_1 \geq 0.$$

(b) $2 + (-1) = 1 \geq 1$ and $2 - (-1) = 3$: feasible. Standard form: $x_1 = 2$, $x_2^+ = 0$, $x_2^- = 1$, $e_1 = 1 - 1 = 0$. The objective is $2 \times 2 - 1 = 3$ (and -3 in the standard form). In fact this is the best plan: the equality gives $x_2 = x_1 - 3$, so $z = 3x_1 - 3$, and the first constraint gives $2x_1 - 3 \geq 1$, that is $x_1 \geq 2$; so $z \geq 3$. Note that the best value of x_2 is negative: the variable really needed to be free.

Q9. $h_1 - h_2 \leq 3$ and $h_2 - h_1 \leq 3$ (Rule 1.7). “At least 3 hours” is impossible: (3, 0) and (0, 3) satisfy it, but their middle point (1.5, 1.5) has a difference of 0.

Q10. (a) $x_1 > x_2$ with whole numbers: $x_1 - x_2 \geq 1$. (b) $x_2 < 16$ with whole numbers: $x_2 \leq 15$. The plan (10, 15) uses 15 cameras, so it is still allowed. Adding a constraint can only remove plans, never add better ones; since the old best plan is still allowed, it is still the best plan.

Appendix: the best plans of the four examples

In this lecture we only *wrote* the models. For your information, here are their best plans, computed with a solver. The drone workshop is solved by hand in Lecture 2.

Example	Best plan
1.1 Drone workshop	(10, 15): 10 drones A and 15 drones B per week, profit 1300 thousand DA. Assembly and testing are fully used; 3 cameras are left. With 61 testing hours: LP optimum (10.5, 14.75) with 1305; best whole-number plan (10, 15) with 1300 (Section 2.7).
1.2 Mixing	About 589.9 kg of M1, 227.3 kg of M2 and 182.7 kg of M3, cost about 229 281 DA; ingredient A is exactly 18% and ingredient B exactly 8%.
1.3 Transport	$W1 \rightarrow S1$: 50, $W1 \rightarrow S2$: 250, $W2 \rightarrow S1$: 150, $W2 \rightarrow S3$: 200; cost 465 000 DA (the plan of Q4).
1.4 Assignment	$R1 \rightarrow T2$, $R2 \rightarrow T1$, $R3 \rightarrow T3$: total time $2 + 6 + 1 = 9$ minutes. The six possible assignments take 9, 10, 14, 16, 20 and 21 minutes.