Introduction to Optimisation

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.

1 An introduction by examples

We introduce mathematical optimisation using examples coming from daily life and other areas of mathematics.

Example 1 (Diet problem)**.** A public servant, Jenny, is responsible for designing a school menu. She has a table (see Table 1) listing servings of six possible foods and their relative nutritional values: calories, protein, and calcium. She also knows how much each serving would cost.

Food	**Energy** (kcal)	**Protein** (g)	**Calcium** (mg)	**Cost** (€)
Chicken	200	30	12	1.5
Eggs	160	15	50	1.0
Milk	160	8	280	0.8
Pork	280	25	120	1.1
Cherry pie	420	4	12	1.6
Vegetables	80	2	8	0.6

Table 1: Food nutritional value and cost, per serving, in the diet problem.

To meet nutritional needs, each child needs 2000 kcal, 55 g of proteins and 800 mg of calcium. Jenny wants to devise a menu satisfying the above levels at the lowest possible cost. (In real life, please do not devise children menus with the main aim of reducing costs!)

A possible approach to the problem is considering six variables $x_1, \ldots, x_6$. The menu will contain x_1 servings of chicken, x_2 servings of eggs, etc., up to x_6 servings of vegetables. Jenny can then easily translate the nutritional requirements into mathematical constraints:

$$\begin{array}{rcrcrcrcrcrcr} 200x_1 & + & 160x_2 & + & 160x_3 & + & 280x_4 & + & 420x_5 & + & 80x_6 & \geq & 2000 \\ 30x_1 & + & 15x_2 & + & 8x_3 & + & 25x_4 & + & 4x_5 & + & 2x_6 & \geq & 55 \\ 12x_1 & + & 50x_2 & + & 280x_3 & + & 120x_4 & + & 12x_5 & + & 8x_6 & \geq & 800. \end{array}$$

She can also represent the cost of a menu using the same variables:

$$\text{cost} = (1.5x_1 + 1.0x_2 + 0.8x_3 + 1.1x_4 + 1.6x_5 + 0.6x_6)€.$$

We can, therefore, precisely describe Jenny's task in mathematical terms:

$$\begin{aligned}
\min \quad & 1.5x_1 + 1.0x_2 + 0.8x_3 + 1.1x_4 + 1.6x_5 + 0.6x_6 && (1)\\
\text{subject to} \quad & 200x_1 + 160x_2 + 160x_3 + 280x_4 + 420x_5 + 80x_6 \geq 2000 && (2)\\
& 30x_1 + 15x_2 + 8x_3 + 25x_4 + 4x_5 + 2x_6 \geq 55 && (3)\\
& 12x_1 + 50x_2 + 280x_3 + 120x_4 + 12x_5 + 8x_6 \geq 800 && (4)\\
& x_1, x_2, x_3, x_4, x_5, x_6 \geq 0. && (5)
\end{aligned}$$

Equation (1) indicates that Jenny is looking for a menu with minimal cost. Equations (2) to (4) define a valid menu, i.e., they require that the menu satisfies the nutritional requirements. Equation (5) makes sure that the values of the variables "make sense" and, in particular, that they cannot be negative (we cannot have a menu with -1 servings of chicken after all).

Although the above description is precise, it will be of little use to Jenny, who wants to find the *solution* to her problem, not just a description. She then remembers that her friend Alan has taken the "Deterministic Models and Optimisation" course at the Barcelona School of Economics. She emails Alan with her problem data and the description (1)–(5), asking for an actual answer. That is to say, she wants to replace variables $x_1, \ldots, x_6$ with non-negative real numbers which satisfy all the nutritional requirements and minimise the cost.

Alan doesn't take long to reply: the optimal menu consists of 6.63 servings of pork (!) and 0.34 servings of cherry pie. The menu certainly meets the *minimum* nutritional requirements: it provides 2000/2000 kcal, 800/800 mg of calcium, and a whopping 167.2/55 g of proteins. Its total cost is 7.84 € per person. The nutritionists, however, have something to object to this menu's variety and healthiness! Jenny quickly realises that her mathematical formulation is not quite precise. In real life, there are more constraints that she must capture.

She quickly amends the model, establishing a maximum number of servings for each type of food:

$$\begin{aligned}
x_1 \leq 3, \quad x_2 \leq 2, \quad x_3 \leq 5,\\
x_4 \leq 2, \quad x_5 \leq 1, \quad x_6 \leq 5.
\end{aligned}$$

After sending the updated model to Alan, she receives the following solution. The optimal menu consists of 1.38 servings of eggs, five servings of milk, two servings of pork, and one serving of cherry pie. It respects the minimum requirements for nutrients because it provides 2000/2000 kcal, 114.6/55 g of proteins, and 1720.8/800 mg of calcium. The cost per person has increased to 9.18 € per person.

Confident in her model's correctness (and Alan's solution skills), Jenny is satisfied: she has found the lowest-cost menu which complies with her constraints.

Exercise 1. Unfortunately, the nutritionists are not as happy as Jenny about her new menu. They argue that the intake of energy, proteins, and calcium should not only meet a minimum level but also not exceed a maximum. In particular, they set the maximum at 2500 kcal, 100 g of proteins, and 1400 mg of calcium. Help Jenny incorporate these requirements into her model, adding new appropriate inequalities.

Example 2 (Lasso regression)**.** Terry must solve a linear regression problem with independent variables $X_1, \ldots, X_p$ and dependent variable y. His model $\hat{f}$ looks as follows:

$$\hat{f}(X_1, \ldots, X_p) = \beta_0 + \beta_1 X_1 + \ldots \beta_p X_p,$$

where $\beta_0, \beta_1, \ldots, \beta_p \in \mathbb{R}$ are the model parameters.

Terry wants to find good parameters for his model but is unsure what "good" means. After looking it up in a statistics book, he convinces himself that reducing the mean squared error (MSE) is an acceptable way of describing how good a model is. The book also suggests that to find the parameters yielding the lowest MSE, he must solve an Ordinary Least Square (OLS) problem.

Given a dataset with n observations, he denotes with $x_{ij} \in \mathbb{R}$ the value of the j-th independent variable and with y_i the value of the dependent variable, in the i-th observation. Solving the OLS problem, then, amounts to finding the optimal solution of the following mathematical optimisation model:

$$\min \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \tag{6}$$

$$\text{subject to} \quad \beta_0, \beta_1, \ldots, \beta_p \in \mathbb{R}. \tag{7}$$

The interpretation of his model is not straightforward. The optimal parameters $\beta_0^*, \ldots, \beta_p^*$ returned by solving the OLS problem (6)–(7) are all non-zero, but Terry is sure that a good part of his independent variables is unrelated to the dependent variable. He decides he wants to perform variable selection (i.e., choose which independent variables to keep in the model) together with linear regression. His book recommends that he uses an approach called Lasso regression. Among the possible formulations of the Lasso, Terry finds the following one:

$$\min \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \tag{8}$$

$$\text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq T \tag{9}$$

$$\beta_0, \beta_1, \ldots, \beta_p \in \mathbb{R}, \tag{10}$$

where $T > 0$ is a parameter that, the book suggests, Terry should determine by ~~trial and error~~ cross-validation.

Example 3 (0–1 Knapsack Problem)**.** A thief is breaking into a flat and quickly realises there is more to take than his backpack can carry. He must then decide which items to steal and which ones to leave to maximise the loot value. He found n interesting objects. Each object i ($i \in \{1, \ldots, n\}$) has a weight $w_i \in \mathbb{N}$ and, once resold on the black market, can earn him a profit $p_i \in \mathbb{N}$. The backpack can only carry up to $C \in \mathbb{N}$ weight units.

In our case, the robber carries a backpack with a capacity of 4000 g, and he finds the objects described in Table 2. He quickly reaches out to his favourite crime consultant, a certain Prof. Moriarty, to determine which objects to pack. Moriartry proposes a mathematical optimisation model with variables $x_1, \ldots, x_n$ (in our case, $n = 7$). These variables, different from those which we have already seen, can only take two values: 0 or 1. We write that $x_i \in \{0, 1\}\ \forall i \in \{1, \ldots, n\}$. The idea of Moriarty is to mark objects which must be packed with variables taking value 1 and objects to discard with variables taking value 0. For example, the assignment $x_1 = x_2 = x_4 = 1,\ x_3 = x_5 = x_6 = x_7 = 0$ will correspond to the instruction: "Steal the computer, the jewellery, and the paining".

Item	**Weight** (g)	**Profit** (€)
Computer	1800	1000
Jewellery	250	3000
Old book	900	2200
Painting	800	2000
Statue	1600	1100
Memorabilia	600	5000
Guitar	2500	1950

Table 2: Weight and profit of the objects in the Knapsack Problem.

To find the *optimal* subset of objects to steal, Moriarty solves the following model:

$$\max \quad \sum_{i=1}^{n} p_i x_i \tag{11}$$

$$\text{subject to} \quad \sum_{i=1}^{n} w_i x_i \leq C \tag{12}$$

$$x_i \in \{0, 1\} \qquad \forall i \in \{1, \ldots, n\}. \tag{13}$$

Equation (11) indicates that the thief wants to maximise the profit of the objects taken. Equation (12) ensures that the sum of the weights of the objects packed does not exceed the backpack's capacity.

After solving the model, Moriarty can finally give the long-awaited order: "Pack the jewellery, the old book, the painting, and the memorabilia. You will be stealing," he continues, "12 200 € of good stuff, and I will get 10% of that for my consultancy… an honest deal, uh?".

2 Mathematical programming models

Wrapping up the three examples above, we now compare the models we have seen and highlight what they share in common and how they are different. In the process, we also introduce the basic terminology which will accompany us during the rest of the course.

The three models below are examples of **mathematical programming models**.

Diet problem (without restrictions on the number of servings)

$$\begin{aligned} \min \quad & 1.5x_1 + 1.0x_2 + 0.8x_3 + 1.1x_4 + 1.6x_5 + 0.6x_6 \\ \text{subject to} \quad & 200x_1 + 160x_2 + 160x_3 + 280x_4 + 420x_5 + 80x_6 \geq 2000 \\ & 30x_1 + 15x_2 + 8x_3 + 25x_4 + 4x_5 + 2x_6 \geq 55 \\ & 12x_1 + 50x_2 + 280x_3 + 120x_4 + 12x_5 + 8x_6 \geq 800 \\ & x_1, x_2, x_3, x_4, x_5, x_6 \geq 0\,. \end{aligned}$$

Ordinary Least Squares problem

$$\begin{aligned} \text{min} \quad & \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \\ \text{subject to} \quad & \beta_0, \beta_1, \ldots, \beta_p \in \mathbb{R} \,. \end{aligned}$$

0–1 Knapsack problem

$$\begin{aligned} \text{max} \quad & \sum_{i=1}^{n} p_i x_i \\ \text{subject to} \quad & \sum_{i=1}^{n} w_i x_i \leq C \\ & x_i \in \{0, 1\} \quad \forall i \in \{1, \ldots, n\} \,. \end{aligned}$$

Here are some of the characteristics of the above models:

1. We use mathematical programming models to optimise something, that is, to either **minimise** or **maximise** a function. The first two problems are minimisation problems, while the third is a maximisation problem. Each model starts with the symbols "min" or "max" to indicate the problem type. These symbols are highlighted in blue in the models.
2. The function we want to minimise or maximise is called the **objective function**.
3. The variables appearing in the objective function and the constraints are simply called the **model's variables**: in the first example they are $x_1, \ldots, x_7$; in the second example they are $\beta_0, \ldots, \beta_p$; in the third example they are $x_1, \ldots, x_n$.
4. The objective function is marked in red in the models. The objective function is **linear** in the first and the third models. In the second example, it is **quadratic** and **convex**, but not linear.
5. All models must define the domain of definition of their variables. In the first model, all variables are non-negative real numbers. In the second case, they are general real numbers. And in the third case, they belong to the discrete set $\{0, 1\}$. The definition of variable domains is highlighted in yellow in the models.

   The variables in the first two models are called **continuous**: they can take an uncountable set of values. In mathematical programming, continuous variables have uncountable subsets of the real numbers as their domain (for example $\mathbb{R}_0^+$ for the first model, and the whole $\mathbb{R}$ for the second model). The variables in the third model are called **discrete**. In general, a discrete variable can take countably many possible values. For example, a variable whose domain is $\mathbb{N}$ is discrete. In our third model, the number of values taken by each x_i is not only discrete but also finite. In particular, each variable can take two values: 0 or 1. In this special case, we call the variables **binary**.

6. Some models impose further restrictions on which combinations of variable values are valid. These restrictions are given through **constraints**, which are highlighted in green. Three parts make up constraints: a left-hand side (LHS), which contains a function of (some of) the variables; a right-hand side (RHS), which is a constant; and either "$\geq$", "$\leq$", or "$=$" between the LHS and the RHS. Constraints using "$\geq$" or "$\leq$" are called **inequalities**, while those using "$=$" are called **equalities**. Examples of constraints, thus, are:

$$x_1 + x_2 - 4x_3 \geq 5$$
$$\sum_{i=1}^{10} |x_i| \leq 3$$
$$-x_1^3 + x_2^2 x_3 = 4.$$

Going back to our three models, we note the following:

- The first model contains three inequalities. The left-hand sides are all linear functions of the variables. In this case, we call the entire constraint **linear**.
- The second model does not have any constraints. Such a model will be called **unconstrained**; by contrast, a model with at least one constraint is called **constrained**.
- The third model contains one linear inequality.

3 Solutions

Exercise 1

The new inequalities are:

$$\begin{array}{rcrcrcrcrcrcr} 200x_1 & + & 160x_2 & + & 160x_3 & + & 280x_4 & + & 420x_5 & + & 80x_6 & \leq & 2500 \\ 30x_1 & + & 15x_2 & + & 8x_3 & + & 25x_4 & + & 4x_5 & + & 2x_6 & \leq & 100 \\ 12x_1 & + & 50x_2 & + & 280x_3 & + & 120x_4 & + & 12x_5 & + & 8x_6 & \leq & 1400. \end{array}$$