TOPIC

Instructions

QUIZ:PROSIT: WHAT'S THE PROBLEM?

(This prosit will be conducted in the context of the project: if necessary, read the statement again before starting)

You now have the proof of your problem's complexity. It's time to find a solution to solve it, a solution that can be applied on a large scale. One thing is for sure, it will be impossible to deliver an optimal solution in polynomial time. The task is becoming more and more complex.

You had made a promise to yourself not to bother her again, but faced with the difficulty, you can't resist calling Agathe to check on her and, of course, to get her opinion on the matter.

Agathe: 'For a small set of cities, I think you can expect to converge to an optimal, if you have enough computation time. For a larger number of cities, it would be nice if your algorithm still delivered a solution, even if it is a suboptimal one. This requires an approach that combines these two objectives, an adaptive approach.'

You: 'Are you thinking of Operations Research?'

Agathe: 'Yes, I am. If you manage to model the problem correctly, you will have access to many generic problem-solving methods.'

You: 'But we already have the model, that's how we were able to prove that the problem was hard.'

Agathe: 'No, I'm talking about a model that matches the algorithm you're using.'

You: While I was reading some literature, I come across systems of equations/inequations. Is that what we're talking about?'

Agathe: 'In case you use the Simplex, that's it. But it depends because OR offers several approaches, and each of them has its own modelling. Take dynamic programming, for example. The optimization problem must be represented as a set of subproblems.'

You: 'Ah, programming, finally! I was getting restless...'

Agathe: 'Sorry to disappoint you, but in this case the term programming doesn't refer to code, it refers to planning'.

You: 'So what do you recommend? The Simplex?'

checked.

You: ‘Okay, I’ll try to create a linear program to see if it does. And if I can’t, then be it dynamic programming...’

Agathe: ‘Not necessarily, there are also metaheuristics.’

You: ‘It’s been a long time since your last tricky word...’

Agathe: ‘It’s worth a look. I’ve heard they’re easier to implement than dynamic programming. And since you have to deliver a prototype in a few days, this is perhaps the least complex approach...’

You: ‘All right, I’ll wait a little longer before I start coding. It’s better that I select the method best suited to my constraints. Thank you for your help! See you soon.’

Or not. You make a promise to yourself (once again) not to bother her for your project.

Resources for students

- Opertaions research - An introduction - EN (Chapters 1, 2, 3.1, 3.2, 3.3, 5, 6, 7)
- An-Introduction to Linear Programming and Game Theory - EN
- Operation Research : - (Linear Programming Model) video - EN
- Linear Programming (LP) With Excel Solver - video EN

EDUCATIONAL GUIDE

Tutoring help

Learning Outcomes

- Operations Research - general info

- Distinguishing between the different classifications of optimization problems: continuous vs discrete, with or without constraints, etc.
- Determining the types of specific problems that can be solved using the methods of operations research
- Describing the computational models used in operations research

- Linear programming

- Interpreting and solving an optimization problem using linear programming

Prosit philosophy and false leads

- The purpose of this Prosit is to introduce Operations Research. As this is a very broad field, students must be able to identify the main categories of approaches used, their limits and constraints. The next prosit will discuss metaheuristics in more detail, meaning that this is the solution that students should have chosen for the prosit outcome and feedback.
- As an introduction, they must detail Linear Programming in Real Numbers (system of linear equations

  /inequations, canonical form and standard form, possible problem-solving methods: graphic, algebraic and simplex). They must understand the simplex philosophy without mastering the details of this algorithm. An example

- In the Workshop, students will also be led to address Linear Programming in Real Numbers, and to see that algorithms capable of solving LPRN problems cannot be used 'as is' for processing ILP (Integer Linear Programming) problems. Therefore, this part allows them to answer the first part of the action plan (determining whether the simplex can be used). Approaches used for solving ILP problems (particularly branching methods) cannot be discussed in detail due to lack of time.
- Moreover, students do not have to create a Linear Program for their problem at this stage, they just have to check if it is possible to model the problem this way.
- The Exercise Series should allow them to practice modelling and solving optimization problems by completing some exercises without involving computer tools.

Descriptif des ressources

Resources in English

- Operations research - An introduction - EN (Chapters 1, 2, 3.1, 3.2, 3.3, 5, 6, 7)
- An-Introduction to Linear Programming and Game Theory - EN
- Operation Research : - (Linear Programming Model) video - EN
- Linear Programming (LP) With Excel Solver - video EN

Original resources in Freanch

\- *Scholarvox literature: A la découverte des Graphes et des algorithmes de graphes, Christian Laforest, EDP SCiences* (chapter 23)

Book on graphs. The specified chapter addresses methods used when problems are hard.

\- *Scholarvox literature: Précis de recherche opérationnelle, Faure et Robert, Dunod* (introduction, chapter 8.1 and 8.2)

Book dedicated to operations research. The introduction explains what operations research is. Chapter 8.1 explains how to apply linear programming from a geometrical point of view. Chapter 8.2 discusses linear programming using the simplex algorithm.

\- *Operations Research models*

Book 'Modèles de Recherche Opérationnelle' by Fabian Bastin. 2010. All models are well detailed; linear, non-linear, dynamic programming, etc.

\- Techniques de l'ingénieur - Linear programming [see 'ressources TI - programmation linéaire.pdf']

Reference resource on linear programming from Techniques de l'ingénieur

\- *Operations Research - modelling example (video)*

Video that presents the fundamentals of OR, also providing a detailed modelling example

Notes for the tutor, tricks an tips

It is important that students understand that the purpose of this sequence is not to directly implement an algorithm, but to decide which type of method should be used for the next sequence.

Moreover, they must meet two important constraints:

- The method must be scalable, i.e. on small-length problems, we can expect it to find an optimal solution, but that it can also handle large-length problems, even if that means finding a suboptimal solution. This is what justifies the use of OR in general.
- It must be fairly easy to set up from a technical point of view. This excludes approaches by Integer Linear Programming from the outset. The Workshop is precisely what illustrates the difference between ILP and LPRN problems, and the fact that the simplex alone cannot be used to solve an ILP problem. Moreover, their resources will often address Linear Programming by going deeper on the explanations (Relaxation, Primal/Dual...). It is important that students understand quickly that they should not go too deep into this subject.

Discovering and clarifying the situation

- *Unknown words:*

modelling the problem, system of equations/inequations, problem-solving methods, operations research, simplex, dynamic programming, subproblems, metaheuristics

- *Context:*

The discussion about NP-Complete and NP-Hard problems made it possible to determine that the processed problem was hard, and that it did not seem feasible to design an algorithm that would produce the optimal solution in polynomial time. We seek advice from Agathe again in order to find an algorithm that could still try to address this problem.

Analyzing the need

The goal is to solve NP-Hard optimization problems by using Operations Research methods, but especially focusing on cases where these problems are of large length.

- *Problem:*

Selecting a method for solving NP-Hard optimization problems, from Operations Research, that could be applied to the maintenance route problem.

- *Constraints:*

The method must work with large-length problems, even if this means terminating it on a suboptimal solution if the computation takes too long

It must be fairly easy to implement

- *Deliverables:*

Design of the algorithm Implementation

Generalization

- - Methods for solving hard optimization problems

Possible solutions

- Can the different calculation methods of OR be applied interchangeably?
- Do you know of any methods for solving equations that could be used to optimise a problem? How could they be used?
- Why do we differentiate between the modelling discussed in the previous prosit and the model used by a method? What does this have to do with the previous question about equations?
- Can we find several solutions to an NP-Hard problem, which are not necessarily optimal?
- What is the relationship of OR with the notion of algorithmic complexity seen in the previous prosit?

Action plan

1. Try to solve the problem with the simplex

- Check that the simplex allows us to search for the optimal solution but can be stopped on a suboptimal solution if the computation time is taking too long (typically, non-polynomial)
- Check if it is possible to represent the problem using a linear program
- Check if the implementation can be done smoothly

2. Regarding other existing methods, including:

check that the method allows us to search for the optimal solution but can be stopped on a suboptimal solution after a certain time

If so:

- Try to represent the problem according to the model used by this method
- Check if the implementation can be done smoothly

3. Select the easiest valid method to implement.
4. Implement the algorithm

Prosit Outcome and Feedback

Tutoring help

Definitions

- ***Operations Research:***

Operations Research is a quantitative approach for making better decisions. It provides tools to streamline, simulate and optimise the architecture and operation of industrial and economic systems. It offers models to analyse complex situations and allows decision-makers to choose efficient and strong solutions.

Operations Research is a discipline that exploits the most operational aspects of mathematics, economics and computer science, being in direct contact with the industry and playing a key role in maintaining competitiveness.

Its contributions are visible everywhere: from the organisation of automotive production lines to the planning of space missions, from the optimization of bank portfolios to assistance with DNA sequencing, but also in everyday life regarding waste recycling, the organisation of school bus services, nurse scheduling or satellite coverage of mobile phones, etc.

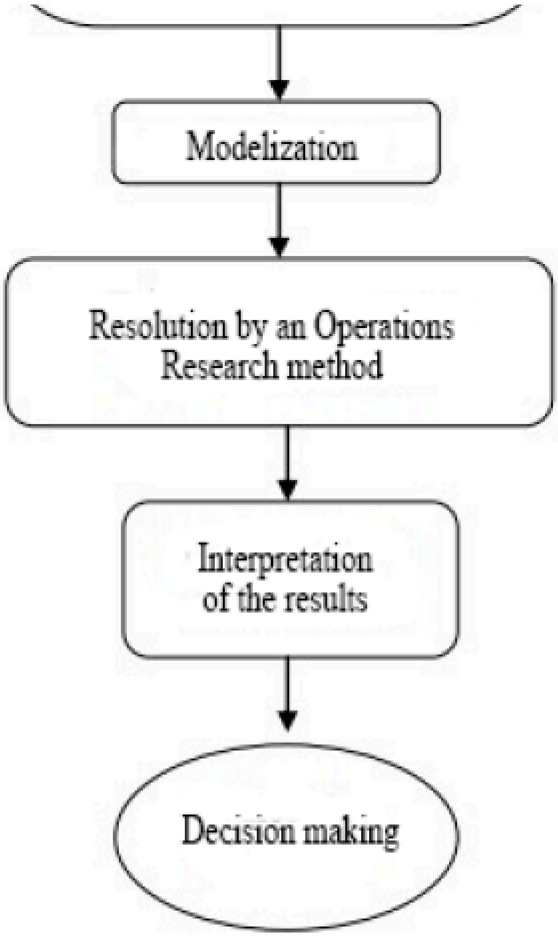


Diagram of the Operations Research process

The use of Operations Research is being increasingly sought after in daily life and at different levels since it offers several advantages and addresses several aspects, such as:

- Business competitiveness improvement,
- Job preservation,
- Access to innovation,
- It proposes the best strategic decisions,
- Provides better resource management, etc.

- ***Formulation of an optimization problem:***

Modelling the problem or converting the text into mathematical equations is not randomly done. It must meet certain conditions and go through several stages: analysis, modelling and the criterion to be optimised:

1. **Analysis**: In this phase, we always start by identifying the data needed to solve the problem, the consumption values or technical data sheets of articles, the quantity of available resources, the transport costs, the production capacity, the earnings generated by each article or the expenses established for carrying out an action. In this phase, we determine the components, challenges and limits of the problem. All this information is essential for problem formulation.

2. **Modelling**: Optimization is always based on mathematical models that are usually simple and partial. To define the system model, we define:

- a- **Decision variables**: they represent the decisions to be made: what has to be done/produced and in what quantity? How much should we buy? How many classrooms does a school need? **The variables represent the**

- b- **Constraints**: conditions expressed as relationships (equations) that the variables must comply with in relation to the available resources and the problem data. These constraints are a n-dimensional space restriction of the solutions to the problem, and this restriction is the space of feasible solutions (which comply with the constraints)

3. **Criterion to be optimised**: It represents the objective or expected goal of the problem. It can either be a minimisation function when dealing with an expense or an investment, or a maximisation function when dealing with a profit or gain. In general, this is referred to as Objective function or fitness function.

Therefore, the goal is to determine, among all value combinations of decision variables that comply with the constraints, the value combinations for which the Objective function is optimal.

Example:

A small company manufactures tables and chairs using two materials: wood and paint, knowing that the production of a table requires 3 m of wood and 4 kg of paint, while the production of a chair requires 2 m of wood and 1 kg of paint. The company's financial resources accommodate a supply of 100 m of wood and 120 kg of paint per week. The products manufactured this way provide a profit of €5 per table and €3 per chair sold.

Question: Formulate the problem!

Solution: we take the data from the example and enter it in a table:

-

	Tables	Chairs	Stock
Wood	3	2	100
Paint	4	1	120
Profit	5	3	

✓ **Variables**

$x1$: is the quantity of tables

$x2$: is the quantity of chairs

✓ **Constraints**

for wood: $3\,x1 + 2\,x2 \leq 100$ (1)

for paint: $4\,x1 + x2 \leq 120$ (2)

We assume that the variables are positive, so we add an additional constraint to the problem:

$x1, x2 \geq 0$ (3)

➔ Constraints $\begin{cases} 3x1 + 2x2 \leq 100 & (1) \\ 4x1 + x2 \leq 120 & (2) \\ x1, x2 \geq 0 & (3) \end{cases}$

✓ **Objective function**

Profit : $z = 5\,x1 + 3\,x2 \longrightarrow$ Objective : $\max z = 5\,x1 + 3\,x2$

- ***Programmation linéaire***

In mathematics, Linear Programming (LP) problems are optimization problems where the Objective function and the constraints are all linear functions of decision variables.

Example (paint production): A company produces interior and exterior paint using two commodities - M1 and M2.

Data:

	Quantity used per metric ton		Quantity available per day
	Outside	Inside	
M1	6	4	24
M2	1	2	6
Profit per metric ton	5	4	

Additional constraints :

- Maximum demand for interior paint: 2 tons/day.

- Interior paint production can only exceed exterior paint production by one ton.

Formulation :

- *Variables*:

  $x1$ = tons of exterior paint produced per day;

- *Constraints*:

$$\begin{cases} 6\,x_1 + 4\,x_2 \;\le 24 \\ x_1 + 2\,x_2 \;\le 6 \\ x_2 \;\le 2 \\ x_2 - \,x_1 \;\le 1 \\ x_1, x_2 \ge 0 \end{cases}$$

- Classification of optimization problems

1. **Continuous linear optimization problems versus discrete linear optimization problems** In some cases, the decision variables are discrete, most often in the form of integers or binary numbers. The optimization problem is said to be discrete, and we are also talking about Integer Linear Programming (ILP). On the contrary, in continuous optimization problems, the variables can take any value, which means that these values are real numbers, and so we are also talking about Linear Programming in Real Numbers (LPRN).

Continuous optimization problems are normally easier to solve. An optimization problem that combines continuous variables and discrete variables is said to be mixed. Discrete or mixed optimization problems are at least NP-Hard, whereas continuous optimization problems are polynomial.

In fact, in the continuous case, the solution space bounded by the constraints is a convex polytope (n-dimensional generalisation of a polygon). This is the property that the Simplex algorithm will exploit to find the optimal solution (see further below).

In the previous example, we are dealing with a LPRN problem.

2. **Mono-objective or multi-objective optimization problems**

Mono-objective problems are characterised by a single Objective function. Multi-objective problems exist when a compromise is to be sought between several conflicting objectives. It might be possible (but not necessarily efficient) to reformulate a multi-objective problem with a single Objective function as a combination of the different objectives, or by transforming objectives into constraints.

In the previous example, we are dealing with a mono-objective problem.

3. **Deterministic or stochastic optimization problems**

Deterministic optimization problems consider that the data is completely known, whereas in stochastic optimization problems this is not the case; for example, a stochastic approach may be relevant in the case where the variables of a problem are the future sales of a product. In this case, uncertainty can be incorporated into the model as random variables.

In the previous example, we are dealing with a deterministic problem.

- ***Standard form & Canonical form of a Linear Program***

Definition 1: A linear program is in standard form when all its constraints are equalities and all its variables are non-negative.

Definition 2: A linear program is in canonical form when all its constraints are inequalities and all its variables are non-negative.

This differentiation is important because some implementations of problem-solving methods (found in some OR libraries) only consider one of these forms.

- ***Methods for solving a LPRN problem***

All methods rely on the fact that the solution space is a convex polytope. This implies that there is at least one vertex of this polytope that represents an optimal solution. This is trivial to observe in the 2-dimensional case (example shown below), quite easy to understand in 3 dimensions (since the constraints and the Objective function are planes), and the generalisation to n dimensions follows the same principle.

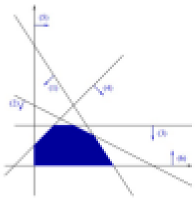
1. **Graphical method**

With the previous example: max $z = 5\,x_1 + 4\,x_2$

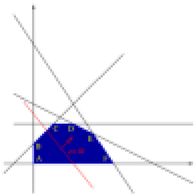
Under the following constraints:

$$\begin{cases} 6\,x_1 + 4\,x_2 \le 24 & (1) \\ x_1 + 2\,x_2 \le 6 & (2) \\ x_2 \le 2 & (3) \\ x_2 - x_1 \le 1 & (4) \\ x_1 \ge 0 & (5) \\ x_2 \ge 0 & (6) \end{cases}$$

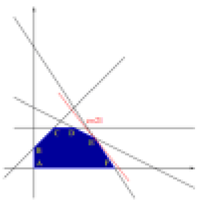
We graphically represent the problem. The two axes represent the two decision variables, and each line corresponds to the equation of a constraint.



The admissible solution space is the polygon ABCDEF. In this space, we look for the values of (x1, x2) that maximize the Objective function value. The set of possible values of the Objective function corresponds to the moveable line whose gradient is (-4, 5) (value derived from the Objective function):

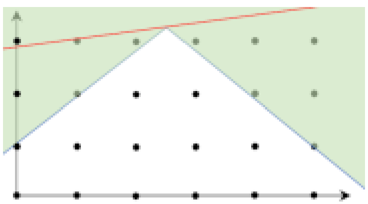


The **Graphic problem-solving** consists in moving the moveable line as far as possible to the right (or to the left if we consider a minimisation problem), until reaching the limits of the polygon:



At this point (the intersection point where the line and the polygon meet), the value of the Objective function is the maximum value for which the decision variables comply with the constraints.

This method helps to illustrate why an Integer Linear Programming problem cannot be solved by the Simplex algorithm:



In the example above, the solution space is in white (bounded by the two blue lines), the dots represent the integer solutions, and the red line represents the Objective function. The solution obtained where the line and the polygon intersect is not integer, being therefore invalid. Moreover, a feasible solution is not obtained even by rounding the decision variables.

2. **The Simplex method**

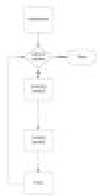
The Simplex is an iterative algorithm for solving a linear programming problem. It consists of starting from an initial vertex of the polytope, and looking for a neighbouring vertex of the current vertex, which improves the objective. This

The general principle of the algorithm is as follows:

- We consider non-basic variables: these are the variables of the system that have (or to which we initially set) a value 0.

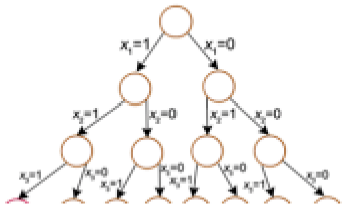
- We consider basic variables: those with a value other than zero.

- Pivoting between the leaving (or outgoing) variable and the entering (or incoming) variable: step of the algorithm which allows to pivot a variable out of the program (out of the base) to put it in the base. The entering variable is chosen according to the variable that improves the fitness function the most. The leaving variable will be the one with the smallest value.



- ***Exact methods for solving an NP-Hard problem***

The main category of methods for solving an NP-Hard optimization problem encompasses the methods by separation/evaluation. It is a set of exact methods (algorithms that find the optimal solution, but not necessarily in polynomial time). Also called Branch & Bound, these methods consist of building an implicit search tree of all existing solutions. This tree will divide the set of solutions into several subsets according to a decision variable, and then split each subset again according to another decision variable, and so on (this is the separation/branch part).



In such a tree (implicit, i.e. that is not stored in memory, being dynamically generated instead), generating the set of leaves is basically generating the set of solutions, including the optimal one. Browsing such a tree is of exponential complexity, but the goal is to remove some branches without having to browse them. Assuming that we already know a feasible solution, and that we are able to compute a lower bound (in the case of a minimisation problem) for all the solutions of a subset. For a given subset, if its lower bound is greater than the value of the current solution, we know it is useless to browse this branch, so we say that the branch is dominated. We therefore decide to ignore this entire branch (this is the evaluation/bound part).

This approach can be used on non-linear problems (see further below), but one of the best-known approaches is applied to ILP: the method by continuous relaxation: We relax (i.e. we temporarily ignore) the integrity constraints (requiring integer decision variables), to make the problem easy to process, and then we derive a lower bound from this problem using the Simplex algorithm. On a two-dimensional LP is easy to check that an optimal solution to a LPRN problem is a lower bound of the optimal solution for the corresponding ILP problem.

Some of its popular variants include:

- Branch & Price: integrity constraints are relaxed at first, then they are reincorporated little by little when exploring the solution tree (this approach is sometimes called column generation because it is basically adding columns to the LP matrix)

- Lagrangian relaxation: hard constraints are removed and integrated into the objective function by penalising said objective function if these constraints are not complied with (this approach is based on the dual problem of the problem considered). This approach is also used to address some non-linear problems (see further below)

Many other variants are designed and proposed on a regular basis (particularly with regard to the order in which the tree is browsed).

- ***Other computational models used in OR***

/evaluation).

1. **Non-linear programming:**

A non-linear program consists of a non-linear Objective function. There are several variants:

- With or without constraints, eventually non-linear ones

- Differentiable or non-differentiable Objective function (the differential is a generalisation of the notion of derivative, and therefore allows optimization approaches such as gradient descent)

It should be noted that even in continuous form, an NLP problem is harder to solve than a LP problem, since the Objective function is non-linear. Therefore, the solution space is no longer a convex polytope, and we cannot apply a method based on the assumption that there is a vertex of this polytope constituting an optimal solution.

Without going into details, the methods for solving such problems are often based on exploring the solution space, where solutions find a local optimum, without guaranteeing that this optimum is global (the notions of local optimum and global optimum will be discussed in more detail during the next class). This exploration is sometimes done by an algorithm similar to the Simplex, but whose application differs from the one considered here. If students have addressed this aspect, it is important to tell the difference.

2. **Dynamic programming:**

Programming is an exact method based on the idea that in order to solve a problem, the smallest subproblems are solved first and the values of these subproblems are kept in a dynamic programming table. We then use these values to compute the value of increasingly larger subproblems, until we obtain the solution to our global problem. This principle seems similar to that of the methods by separation/evaluation, but the difference here is that we do not consider a tree, as the subsolutions common to several solutions are computed only once.

3. **Metaheuristics (discussed in detail in the next Prosit)**

A metaheuristic is an algorithmic method capable of guiding and steering the search process in a solution space, which is often very large and contains regions packed with optimal solutions. Making this method abstract and more generic leads to a wide use for different application fields. This is where metaheuristics differ from the exact methods, which do guarantee that the problem will be solved, but at the expense of prohibitive computation times.
Validation questions

- What kind of problems does OR tackle?

NP-Hard optimization problems of large length.

- Give examples of specific problems that can be solved with the methods of OR

Event planning under resource constraints (timetabling problem)

Task scheduling (supply chain, assembly line...)

Allocation of production tasks to industrial equipment

Transport problems (travelling salesman, vehicle routes...)

Optimization of resource use (knapsack)

etc.

- What are the different modelling/problem-solving methods in OR?

Linear Programming (discrete or continuous)

Non-linear Programming

Constraints Programming

etc.

- What are the problem-solving steps in LP?

The different steps are summarised in the diagram below

- Can we model the problem using Linear Programming? Can we solve it using the Simplex algorithm?

Yes, we can model it using a Linear Program, but it will be discrete, and the Simplex algorithm will not be enough to solve it

- Does the Simplex algorithm solve an Integer Linear Programming problem?

No, the Simplex only solves Linear Programming problems in Real Numbers. In the discrete case, the Simplex can be used by more complex methods (branching methods) to build a solution.

- How do you know if an optimization problem can be solved using dynamic programming?

The optimal solution to a problem is composed of optimal solutions that contain subproblems

To be adapted according to the processed problem (selection of additional constraints):

The (basic) problem of delivery routes matches the metric TSP problem. This problem can be modelled as an ILP problem. The Simplex alone does not make it possible to build an optimal (or even admissible) solution.

The exact branching-type methods would be a solution based on LP. However, they are excluded due to their implementation complexity because there is no time to set up such an approach in the context of this prosit.

Constraints Programming is also a solution, which is often used, but again its approach is too complex for this prosit.

Given that Dynamic Programming is a more affordable solution, it is possible that students have chosen it. But it is still a little too complex.

Metaheuristics are therefore the preferred solution.

EXERCISE SERIES

TOPIC

Resources for students

Exercise series

- Part 1:

  Statement [doc] - FR

  Correction [doc] - FR

- Part 2:

  Statement [zip]

WORKSHOP

TOPIC

Instructions

QUIZ

Énoncé du Workshop [zip] (Notebook Jupyter)

EDUCATIONAL GUIDE

Standard corrections

Workshop version tuteur [zip]