Norges Teknisk-Naturvitenskapelige Universitet

TPK4186 - Advanced Tools for Performance Engineering
Spring 2023

Assignment 4: Construction Project

Prepared by Antoine Rauzy

Contents

1 Introduction

1.1 Presentation of the Problem

PERT Diagrams Construction projects consist of separated tasks. For instance, excavating the place where the building will stand, building the foundations, building the walls and the floors, building the roof. . . Some tasks must be finished before other can start. For instance, excavating must be completed before on can start building the foundations. Each task has a duration. This duration is subject to uncertainties. The total duration of a project is thus also subject to uncertainties.

PERT diagrams are used to study what is the expected duration of a project, how much the project can be delayed, which tasks are on a critical path, i.e. cannot be delayed without delaying the whole project, and so on.

As an illustration, Table 1 describes the tasks of the construction of a warehouse.

Figure 1 shows the corresponding PERT diagram.

There are actually two types of tasks: actual tasks, here labeled from A to J, and gates, here Start and End, which represent important steps of the project. In our case, gates represent only the start and the completion of the project, but in general intermediate gates are added as well. We shall come back on this point. From now, we shall say “task” for both tasks and gates.

Saying that the task X is a predecessor of the task Y is saying that X must be completed before Y starts. In the diagram, this is represented by an arrow going from the node representing the task X to the node representing the task Y. Not that the diagram is a directed acyclic graph, i.e. it contains no circuit.

Types	Codes	Descriptions	Duration	Predecessors
Gate	Start			
Task	A	Agreement of the owner	(3, 4, 6)	Start
Task	B	Preparation of the ground	(1, 2, 4)	Start
Task	C	Order of materials	(1, 1, 3)	A
Task	D	Excavation	(1, 1, 3)	A, B
Task	E	Order of doors and windows	(1, 2, 4)	A
Task	F	Delivery of materials	(1, 2, 4)	C
Task	G	Concrete foundations	(1, 2, 4)	D, F
Task	H	Delivery of doors and windows	(8, 10, 12)	E
Task	J	Frame, roofs, walls	(3, 4, 6)	G
Task	K	Installation of doors and windows	(1, 1, 3)	H, J
Gate	End			J, H

Table 1: Construction of a warehouse

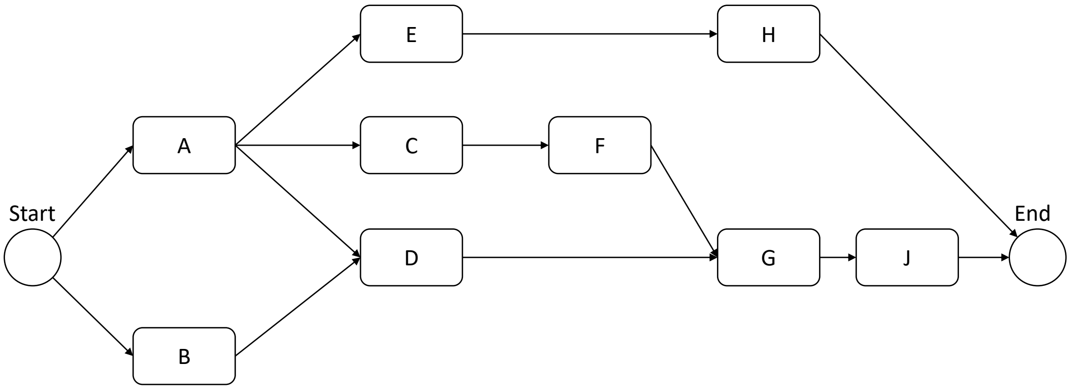

Figure 1: PERT diagram for the warehouse project

As already said, each task has a duration. This duration is however uncertain. In the sequel, we shall assume that durations obey by triangular distributions. Triangular distributions are characterized by 3 parameters: a minimum value, an expected value (also called mode), and a maximum value (see the book for further explanations). The Python library `random` provides a method to draw duration at random according to triangular distributions.

Early and Late Dates Once the duration of each task determined, four dates are calculated for each task:

– The early start date.

– The early completion date.

– The late start date.

– The late completion date.

The early start date of a task is the ~~minimum~~ MAX of the early completion dates of its predecessors, and 0 if the task has no predecessor. The early completion date of a task, is its early start date plus its duration. There is thus a very simple algorithm to determine the early dates:

1. Start with the list of all tasks.

Table 2: Early and late dates for the warehouse project

Task	Start	A	B	C	D	E	F	G	H	J	K	End
Duration	0	4	2	1	2	2	2	2	10	4	1	0
Early start date	0	0	0	4	4	4	5	7	6	9	16	17
Early completion date	0	4	2	5	5	6	7	9	16	13	17	17
Late start date	0	0	7	7	9	4	8	10	6	12	16	17
Late completion date	0	4	9	8	10	6	10	12	16	16	17	17

2. While it remains a task in the list,
    - Find a task in the list whose all predecessors are not in the list.
    - Remove this task from the list.
    - Calculate the early dates for this task.

The minimum duration of the project is the maximum, over its tasks, of the early completion dates.

Early dates are thus calculated by propagating values from the source nodes (Start in our example) to the target node (End in our example). Late dates are calculated in the reverse way.

The late completion date of a task is the ~~maximum~~ minimum of the late start dates of its successors, and the project duration if it has no successor. The late start date of a task is its late completion date minus its duration. The algorithm to calculate late dates is as follows.

1. Start with the list of all tasks.
2. While it remains a task in the list,
    - Find a task in the list whose all successors are not in the list.
    - Remove this task from the list.
    - Calculate the late dates for this task.

Table 2 gives the early and late dates for the warehouse project, assuming that the durations of tasks are their expected duration.

A task is critical if it cannot be delayed without delaying the whole project, i.e. if its early and late start dates are equal. in our example, tasks A, E, H and K are critical.

Machine Learning Assuming a company builds many warehouses similar to the one described in the previous paragraphs, this company may want to detect as early as possible when a project is drifting.

The idea is then to set up a intermediate gate after a the first tasks are completed. All the tasks located upstream this gate belongs to the first phase of the project, while all the tasks located downstream this gate belong to the second phase of the project. We can then observe the completion dates of the first phase, and try to predict whether the completion date of the project will be delayed. To do so, we can apply machine learning techniques.

In this assignment, we shall use PERT diagrams to generate the learning set and the test set of classification and regression algorithms.

1.2 Requirements

The objective of this assignment is to show your Pythonic skills.

Here follows a number of requirements.

1. You must provide your program together with a small document explaining how it is organized, what it is doing (which functionalities are implemented) and reporting experiments you have performed with it.
2. The assignment can be made either individually or in group (maximum 3 persons, preferably 2).
3. Assuming you are Esben Askelad, all the files you deliver must be included in a zip archive named:

   `TPK4186 - 2023 - Assignment 4 - Esben Askelad.zip`

   The deliverable of the assignment is this zip archive. Please recall your names in addition to the group name both in the header of the program and in the accompanying document.
4. The quality of a program is judged along three criteria: its completeness, its correctness and its maintainability:
   - A program is complete if it provides all functionalities demanded by the client. Some functionalities are however more important than other. You must first concentrate on the main functionalities, then develop the "nice-to-have" ones.
   - A program is correct if it is bug free. To ensure that your program is correct, you must test it extensively. Design tests before writing the first line of code. There is no such a thing than a program or a functionality that works "most of the time". Either it works, or not. If you are not able to make a functionality work, do not deliver it.
   - A program is maintainable if it is well presented, if the identifiers are significant, and so on. But before all, a program is maintainable if it well organized and as modular as possible. Separate the concerns.

2 Tasks

The assignment consists in the following tasks.

2.1 PERT Diagrams

The first step consists in designing (and testing) data structures to implement PERT diagrams.

Task 1. Design data structures to encode PERT diagrams.

Note that you will need for each task, the list of predecessors of this tasks as well as the list of its successors.

Task 2. Design a loader to load projects from Excel spreadsheets. Two examples of spreadsheets are joined: `Warehouse.xlsx` and `Villa.xlsx`.

Design a printer to print projects, including the list of successors of each task, their early and late dates, and their criticality. Use this printer to test both the data structures you designed in Task 1. and your loader.

Task 3. Design a calculator to calculate the early and late dates of each task of a project. Test your calculator on both examples `Warehouse.xlsx` and `Villa.xlsx`.

Design functions to calculate the shortest, the expected and the longest duration of a project.

2.2 Machine Learning

We shall now make some statistics about the durations of the project `Villa.xlsx`. This implies drawing at random the durations of the tasks. As of now, durations of tasks are assumed to obey triangular distributions. Moreover, they are independent one another.

We shall break this independence by introducing a risk factor r. r can take the following values: 0.8, 1.0, 1.2 and 1.4. This value is fixed, once for all, and is the same for all the tasks of the project.

Now, consider a task with minimum duration a, an expected duration e, and maximum duration b. An new expected duration e is calculated as $e' = e \times r$. Indeed, if e' is less than a, then we set e' as a. Symmetrically, if e' is bigger than b, then we set e' as b. Finally, the duration is drawn at random according the triangular distribution with minimum a, maximum b and expected value e'.

We shall also classify the projects into 3 categories:

- Successful, if the actual duration of the project does not exceed its expected duration (with a risk factor 1.0) by more than 5%.
- Acceptable, if the actual duration of the project stands between 105% and 115% of its expected duration (with a risk factor 1.0).
- Failed, if the actual duration of the project exceeds its expected duration (with a risk factor 1.0) by more than 15%.

Task 4. Draw at random a sample 1000 of values of durations, for each value of the risk factor, according to the above principle. Calculate the project duration for each project of the sample. Perform statistics on these durations (minimum, maximum, mean, standard-deviation, deciles), as well as on the numbers of successful, acceptable and failed projects.

You shall now add an intermediate gate to separate the project into two phases. You can propose several positions for this gate.

You will then generate a sample of instances on which machine learning techniques techniques can be applied. The attributes of the instances are the early completion dates of each tasks upstream the gate. The value of the instancse are the class of the project if a classification method is applied, the actual duration of the project if a regression method is applied.

The sample of instances should mix instances with different values of the risk factor. Concretely, you can draw one of the four possible value at uniformly random (using, for instance, the method `choice` of the package `random`).

Task 5. Draw at random a sample of instances according to the above principle in order to apply classification methods. Apply at least 3 different classification methods. Use 80% of the sample as a learning set and the remaining 20% as a test set. Report and discuss the results.

Task 6. Draw at random a sample of instances according to the above principle in order to apply regression methods. Apply at least 3 different regression methods. Use 80% of the sample as a learning set and the remaining 20% as a test set. Report and discuss the results.