Challenge ROADEF / EURO 2020
Grid operation-based outage maintenance planning
Rules

Manuel Ruiz, Pascal Tournebise and Patrick Panciatici
RTE

April 14, 2020

Contents

1 General rules

The ROADEF/EURO 2020 challenge is dedicated to grid operation-based outage maintenance planning in collaboration with Réseau de Transport d'Électricité (Electricity Transmission Network), usually known as RTE). The problem description, instances and checker are proposed by RTE. Participants compete to propose the best algorithmic solutions to solve the given problem. Informations about the challenge are available on the challenge website. The participants are welcomed to provide feedbacks (questions, doubts, errors, ...) to the organizers concerning the problem. They must submit them on the forum available in the challenge website. Questions asked by email might be ignored by the organization. The organizers try to answer to the participants as fast and accurately as possible.

The rules about team composition are given below:

- A team can be composed of any number of persons
- One person cannot be a member of more than one team
- Each team can submit only one program to solve the problem
- A junior team must include only student members (students enrolled in a Licence, Master's or Doctorate program).
- A team is not considered as a junior team if one or more of its members has defended a Ph.D. on December 31st, 2019.
- Persons employed by RTE or any company of the EDF group cannot participate
- The members of the organization committee cannot participate, but members of their respective laboratories can. In such cases, the organizers do not share with them any information related to the challenge unless it has been previously communicated to all participants.
- Anyone having signed a contract with RTE to work on topics directly related to the subject of the challenge cannot participate.
- To participate in open source with accepted third dependancy category, source code must be available on an open platform (GitHub,...) at the end of the competition, and third party dependancy should be limited to mathematical programming solvers. Competitors should ask if the sofwtare is considered as an acceptable mathematical programming solver, the list will be updated in the challenge rules document.

2 Organization

This section describes the general steps of the challenge.

2.1 Roadmap

The challenge starts on April 1st and final results wil be annonced on February 2022 at ROADEF 2020. The detailed roadmap is available at https://www.roadef.org/challenge/2020/en/calendrier.php.

The teams can participate to the sprint and qualification phase independently to each other. After the qualification phase, only qualified teams defined by the organizers will be allowed to participate to the final phase. Dates might be extended and if it happens, the organizers will notify the participants.

2.2 Prices

The total amount of prizes is 55 k€, divided in several categories as explained at https://www.roadef.org/challenge/2020/en/prix.php

2.3 Datasets

Three datasets composed of industrial problem instances are provided to participants during the challenge:

- Dataset A of 15 instances at the launch of the challenge.
- Dataset B of 15 instances available after the end of the qualification phase.
- Dataset X of 15 hidden instances that will be used to rank the candidates for the final phase and available after the end of the challenge.

Dataset A contains small/medium size instances while datasets B and X contain regular size instances. The maximum number of intervention is 1000. The time horizon may contain at most 365 time stamps. The number of ressources is lesser or equal to 15. The maximum number of risk distribution is 600.

2.4 Evaluation

During the sprint phase, teams are evaluated according to their submitted solution files. At the deadline of the qualification and final phases, each team must send its computer program (see Section 2.6), solution files and a short document (2 pages max. for the qualification and 5 pages max. for the final phase). This document must describe the idea of the solving method, the characteristics of the computer used to find solutions and the computational results table. The dataset A will be used to compare teams during the sprint phase and at the end of the qualification phase, datasets B and X at the end of the final phase.

The computer program of each team will be executed with a time limit of fifteen minutes and another one of one hour and a half. Let Z be a problem instance from a dataset and the solution value for instance Z within fifteen minutes (resp. one hour and a half) is $V_{15}(Z)$ (resp. $V_{90}(Z)$). The following function is used to compute the score $V(Z)$ of a team for instance Z:

$$V(Z) = 0.8V_{15}(Z) + 0.2V_{90}(Z)$$

The weight function between the two solutions is favorable to the fifteen minutes time limit since it is the most frequent used case. Note that if a team fails to provide a solution within one of the time limits for an instance Z or if the retained solution is considered as not valid by the solution checker, the result of a very naive greedy heuristic will be used in the score computation of the team.

2.5 Ranking method

2.5.1 Definition

At the end of each phase, teams are ranked according to their solution values. The used ranking function is the following.

Each team earns points for every instances, depending on their result. We denote by score the number of points earned by a team, whereas result denote the objective value obtained by the team. The winner is the team that maximize the sum of its scores over all instances.

Let m be the total number of instances in a dataset. Let y_{ij} be the result of team i for instance j computed according to evaluation function from Section 2.4. Let $n_{better(ij)}$ be the number of teams with result strictly better than the result of team i on instance j :

$$n_{better(ij)} = \text{Card}(\{k \neq i \text{ s.t. } y_{kj} < y_{ij}\}).$$

Let R be the maximal score that a team can earn form one instance. During both qualification and final phases, R will be equal to 10.

Let p_{ij} be the number of points scored (*i.e.* the score) by team i for the instance j. It is defined by

$$p_{ij} = \begin{cases} \max\{0, R - n_{better(ij)}\} & \text{if feasible,} \\ 0 & \text{if unfeasible or crash.} \end{cases} \tag{1}$$

The global score of a team, denoted as $score(i)$ is defined by :

$$score(i) = \sum_{j \in [1,m]} p_{ij}.$$

2.5.2 Example

The following illustrates how the top team is identified. Consider the following example with $m = 6$ instances of a minimization problem, 7 teams and $R = 5$.

Instance	1	2	3	4	5	6
Team 1	34	35	42	32	10	12
Team 2	32	24	44	33	13	15
Team 3	33	36	30	12	10	17
Team 4	36	32	46	32	12	13
Team 5	37	30	43	29	9	4
Team 6	68	29	41	55	10	5
Team 7	39	30	43	58	10	4

Table 1: Values of y_{ij}

The values $n_{better(ij)}$ are the following:

Instance	1	2	3	4	5	6
Team 1	2	5	2	2	1	3
Team 2	0	0	5	4	6	5
Team 3	1	6	0	0	1	6
Team 4	3	4	6	2	5	4
Team 5	4	2	3	1	0	0
Team 6	6	1	1	5	1	2
Team 7	5	2	3	6	1	0

Table 2: Values of $n_{better(ij)}$

The score are the following:

Instance	1	2	3	4	5	6
Team 1	3	0	3	3	4	2
Team 2	5	5	0	1	0	0
Team 3	4	0	5	5	4	0
Team 4	2	1	0	3	0	1
Team 5	1	3	2	4	5	5
Team 6	0	4	4	0	4	3
Team 7	0	3	2	0	4	5

Table 3: Values of score p_{ij}, with $R = 5$

Finally, the global score are:

Team 1	15
Team 2	11
Team 3	18
Team 4	7
Team 5	20
Team 6	15
Team 7	14

Table 4: Values of $score(i)$, with $R = 5$

In this example, Team 5 would be the winner.

2.6 Computing and software limitations

To make a fair comparison of methods developed by different participating teams, each of them has to deliver for qualification and final phases an executable code. This program must take as input a problem instance from one of the datasets according to format from Section 3.1.1. It must return an output solution file using standards defined in Section 3.1.2. For each instance, the organizers will run ONE trial with fixed seed chosen at random. The same seed will be used for each team and each instance. Using the seed value passed as a parameter to the executable program is under the responsibility of participants to support the repeatability of experiments/evaluations, particularly, if their solution methods are based on probabilistic frameworks or components. Programs will be evaluated only ONCE, never TWICE. If variability occurs, the organizers can not be responsible from potential bad behavior on this single run. The team's submissions will be run by the organizers, they should thus give support to the organizers in the process of running their algorithms.

The computer that should be used to evaluate programs of candidates is a Linux OS machine with 2 CPU, 16GB of RAM. The following list of ILP solvers allowed during the challenge:

- CPLEX
- Gurobi
- LocalSolver

3 Files format and description

3.1 Files format

3.1.1 Input

Each instance is written in a single json file (details are available at http://json.org/) whose elements are :

- Ressources : see below
- Seasons : see below
- Exclusions : see below
- T : see below
- Scenarios_number : see below
- Alpha : the α parameter used in the objective fonction
- Quantile : the τ parameter defining the quantile used to compute the expected excess indicator of the objective function
- ComputationTime : allowed time in minutes for computation

More details on specific elements are described in the following figures.

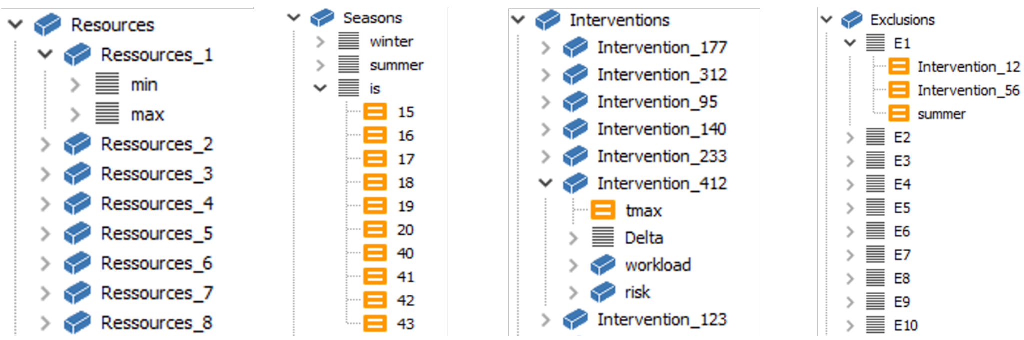

Figure 1: Intermediary level of a json file

Please note that the exclusions are not directly stored as (i_1, i_2, t). They are stored by season, and the link between seasons and time steps are given in the "Seasons" subobject.

In "Resources", "min" and "max" are ordered lists of length T.

In "Scenarios_number", $|S_t|$ values are written in a list of length T.

In "Interventions", "Delta" is an ordered list of length T. "tmax" is a pre-computed value corresponding to the latest possible starting time for a given intervention.

Figure 2 represents the resources workload and risk values of a given intervention. When accessing a certain workload $r_{i,t'}^{c,t}$ (in the above example i = Intervention_308 and c = Ressources_2), t

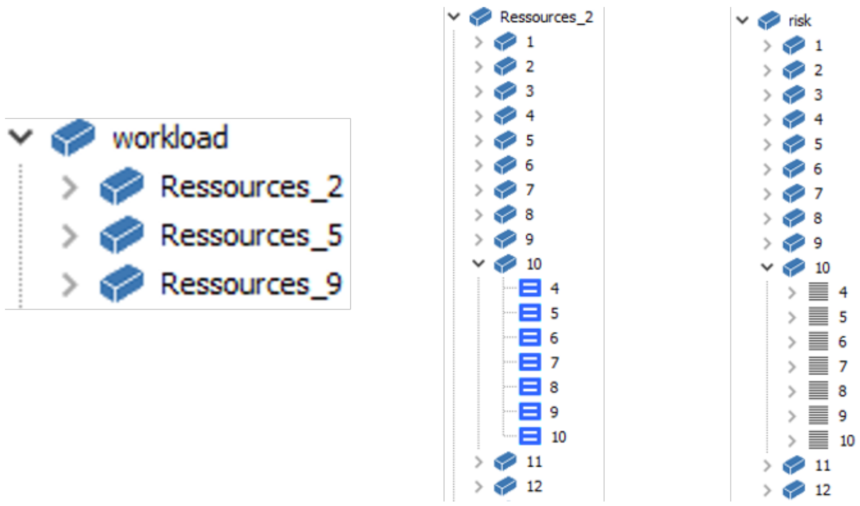

Figure 2: Bottom level of a json file

value has to be given before t'. This means that the expanded part of "Ressources_2" in 2 goes for $t = 10$ and $t' = \{4, ..., 10\}$.

"risk" subobject works similarly except that the end of the tree is an ordered list of size $|S_t|$.

Only relevant values are written down in the json file. In order to limit size file, irrelevant values (such as resources workload or risk for $t > tmax$) are not accessible. They can be considered as equal to 0.

3.1.2 Output

The expected output is a text file (.txt) with $|I|$ lines. Each line includes the intervention name and its starting time, separated by a space. An example is as follow:

Intervention_153 6
Intervention_12 25
Intervention_2 3
...

Note that the interventions can be given in any order.

3.2 Checker

A checker is provided. It is piece of code written in Python and available in github at `https://github.com/rte-france/challenge-roadef-2020.git`

3.3 Small example

Below we provide an example of a small instance and a feasible (but not necessarily optimal) planning. We show the planning is feasible by explicitly computing all constraints.

Sets

- $H = \{1, 2, 3\}$
- $C = \{c_1\}$
- $I = \{I_1, I_2, I_3\}$
- $S_1 = S_2 = S_3 = \{s_1, s_2, s_3\}$
- $Exc = \{(E, 1), (E, 2), (E, 3)\}$ with $E = (I_2, I_3)$

Parameters

	$i = I_1$	$i = I_2$	$i = I_3$
t=1	3	1	1
t=2	3	1	1
t=3	2	1	2

Table 5: $\Delta_{i,t}$ values

	$c = c_1$
t=1	10
t=2	0
t=3	6

	$c = c_1$
t=1	49
t=2	23
t=3	15

Table 6: l_t^c values (on the left) and u_t^c values (on the right)

	$t = 1$	$t = 2$	$t = 3$
$t^{'} = 1$	31	0	8
$t^{'} = 2$	/	/	/
$t^{'} = 3$	/	/	/

	$t = 1$	$t = 2$	$t = 3$
$t^{'} = 1$	14	0	0
$t^{'} = 2$	0	14	0
$t^{'} = 3$	0	0	14

	$t = 1$	$t = 2$	$t = 3$
$t^{'} = 1$	5	0	0
$t^{'} = 2$	0	5	0
$t^{'} = 3$	/	/	/

Table 7: $r_{i,t^{'}}^{c_1,t}$ relevant values (from left to right: I_1, I_2, I_3)

Resource workload values for certain $t^{'} \in H$ are not given because the corresponding interventions cannot start at this time (otherwise they would exceed the time horizon).

Solution example

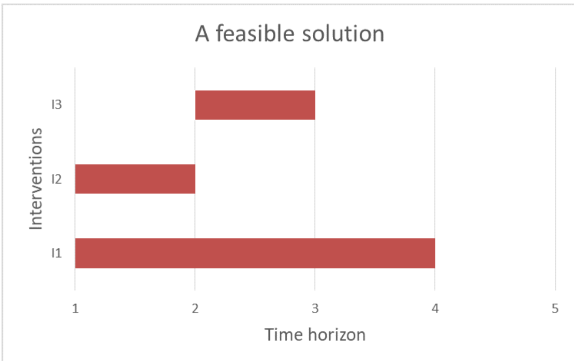

Figure 3: Example of a feasible solution

Feasibility

Interventions are contiguous, and terminate before the end of the schedule. They are planned once, and only at the beginning of a period.

The resource workload at $t = 1$ comes from I_2 and from I_1. The resource workload due to I_2 is 14, and 31 for I_1. Therefore the overall resource workload is 45. As we have $10 \leq 45 \leq 49$, the resource constraint at $t = 1$ holds.

At $t = 2$, the resource workload comes from I_3 (5) and from I_1 (0). It is likely a day off when I_3 can be carried out but not I_1. As $0 \leq 5 \leq 23$, the resource constraint at $t = 2$ holds.

At $t = 3$, the resource workload comes from I_1 only (10). As $6 \leq 10 \leq 12$, the resource constraint at $t = 3$ holds.

I_2 and I_3 are correctly excluded.

Objective

In order to compute the objective, we also need the following parameters.

	$t=1$	$t=2$	$t=3$
$t'=1$	7	1	1
$t'=2$	/	/	/
$t'=3$	/	/	/

	$t=1$	$t=2$	$t=3$
$t'=1$	4	10	4
$t'=2$	/	/	/
$t'=3$	/	/	/

	$t=1$	$t=2$	$t=3$
$t'=1$	8	10	4
$t'=2$	/	/	/
$t'=3$	/	/	/

Table 8: Risk values for I_1 $risk^{s,t}_{I_1,t'}$ (from left to right: s_1, s_2, s_3)

	$t=1$	$t=2$	$t=3$
$t'=1$	5	0	0
$t'=2$	0	5	0
$t'=3$	0	0	5

	$t=1$	$t=2$	$t=3$
$t'=1$	4	0	0
$t'=2$	0	4	0
$t'=3$	0	0	4

	$t=1$	$t=2$	$t=3$
$t'=1$	5	0	0
$t'=2$	0	5	0
$t'=3$	0	0	5

Table 9: Risk values for I_2 $risk^{s,t}_{I_2,t'}$ (from left to right: s_1, s_2, s_3)

	$t=1$	$t=2$	$t=3$
$t'=1$	4	0	0
$t'=2$	0	3	0
$t'=3$	/	/	/

	$t=1$	$t=2$	$t=3$
$t'=1$	8	0	0
$t'=2$	0	8	0
$t'=3$	/	/	/

	$t=1$	$t=2$	$t=3$
$t'=1$	2	0	0
$t'=2$	0	1	0
$t'=3$	/	/	/

Table 10: Risk values for I_3 $risk^{s,t}_{I_3,t'}$ (from left to right: s_1, s_2, s_3)

Let $\tau = 0.5$. The planning risk at different time steps can then be computed:

	$t=1$	$t=2$	$t=3$
s_1	12	4	1
s_2	8	18	4
s_3	13	11	4
Mean	11	11	3
$Q_{0.5}$	12	11	4
$Ecost_{0.5}$	1	0	1

Table 11: $risk^s_{tot}(t)$ values, mean, $Q_{0.5}$ and $Ecost_{0.5}$

For instance $risk_{tot}^{s_1}(1) = risk_{I_1,t'=1}^{s_1,1} + risk_{I_2,t'=1}^{s_1,1} = 7 + 5 = 12$ and $risk_{tot}^{s_3}(2) = risk_{I_1,t'=1}^{s_3,2} + risk_{I_3,t'=2}^{s_3,2} = 10 + 8 = 18$.

Consequently the mean cost value is $obj_1 = \frac{11+11+3}{3} = 8.33$ and the expected excess is $obj_2 = \frac{1+0+1}{3} = 0.66$.

Objective (2)

Let's do another example to show that, generally speaking, we do not have $S_1 = S_2 = S_3$. Consider for instance $S_1 = \{s_1\}$, $S_2 = \{s_2\}$ and $S_3 = \{s_3, s_4\}$.

	$t = 1$
$t' = 1$	7
$t' = 2$	/
$t' = 3$	/

	$t = 2$
$t' = 1$	4
$t' = 2$	/
$t' = 3$	/

	$t = 3$
$t' = 1$	2
$t' = 2$	/
$t' = 3$	/

	$t = 3$
$t' = 1$	20
$t' = 2$	/
$t' = 3$	/

Table 12: Risk values for I_1 $risk_{I_1,t'}^{s,t}$ (from left to right: s_1, s_2, s_3, s_4)

	$t = 1$
$t' = 1$	8
$t' = 2$	0
$t' = 3$	0

	$t = 2$
$t' = 1$	0
$t' = 2$	5
$t' = 3$	0

	$t = 3$
$t' = 1$	0
$t' = 2$	0
$t' = 3$	6

	$t = 3$
$t' = 1$	0
$t' = 2$	0
$t' = 3$	8

Table 13: Risk values for I_2 $risk_{I_2,t'}^{s,t}$ (from left to right: s_1, s_2, s_3, s_4)

	$t = 1$
$t' = 1$	2
$t' = 2$	0
$t' = 3$	/

	$t = 2$
$t' = 1$	0
$t' = 2$	6
$t' = 3$	/

	$t = 3$
$t' = 1$	0
$t' = 2$	0
$t' = 3$	/

	$t = 3$
$t' = 1$	0
$t' = 2$	0
$t' = 3$	/

Table 14: Risk values for I_3 $risk_{I_3,t'}^{s,t}$ (from left to right: s_1, s_2, s_3, s_4)

	$t = 1$	$t = 2$	$t = 3$
s_1	15	/	/
s_2	/	10	/
s_3	/	/	2
s_4	/	/	20
Mean	15	10	11
$Q_{0.5}$	15	10	2
$Ecost_{0.5}$	0	0	0

Table 15: $risk_{tot}^{s}(t)$ values, mean, $Q_{0.5}$ and $Ecost_{0.5}$

In this case $obj_1 = \frac{15+10+11}{3} = 12$ and $obj_2 = \frac{0+0+0}{3} = 0$.

4 Discussion and updates

To help participants managing their project, the organizers try to communicate immediately changes that might occurs during the challenge by updating the dedicated github repository https://github.com/rte-france/challenge-roadef-2020.git. This includes clarification of the problem description or change in problem instances for example. The organizers try to avoid making such changes unless they find it necessary for the challenge.

The organizers reserve the right to disqualify any participant or team from the competition at any time if the participant is considered to have worked outside the spirit of the competition rules.

5 Intellectual property

1. Participants have intellectual property on their computer programs developed during the challenge. RTE and any third party may use information provided by the participants through technical reports, scientific papers and oral presentations, but cannot use a computer program without the agreement of the team who wrote this program.

2. Participants to the challenge cannot claim to have a partnership or a contract with RTE, even if they win the challenge. They can only claim to be participants (respectively qualified / winner) if it is the case.

3. RTE may (but has taken no engagement to) sign contracts with some participants after the challenge. Any such contract would be independent of the challenge.