# Challenge ROADEF / EURO 2022
# Truck Loading
# Rules

Alain Nguyen, Mohamed-Amine Khatouf and Christian Serrano
Renault Group
Version 1.0

July 4, 2022

## Contents

# 1 General rules

The ROADEF/EURO 2022 challenge is dedicated to truck loading at Renault. The problem description, instances and checker are proposed by Renault. 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 welcome to provide feedbacks (questions, doubts, errors, ...) to the organizers concerning the problem. They must ask them on the forum available in the challenge website (issues Github). The organizers try to answer to the participants accurately as fast 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 Undergraduate, Graduate or PhD 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, 2022.
- Persons employed by Renault or any company of the Renault 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 Renault to work on topics directly related to the subject of the challenge cannot participate. Participants to the ESICUP challenge 2015 on Renault's container loading problem are allowed to participate. The topic of the present challenge is somewhat different from that of ESICUP challenge.

# 2 Organization

This section describes the general steps of the challenge.

## 2.1 Roadmap

The challenge starts on July 4th and results will be announced at ROADEF 2023 conference, IFORS 2023 conference and ROADEF 2024. The detailed roadmap is available at `https://www.roadef.org/challenge/2022/en/calendrier.php`.

The teams can participate to the sprint and qualification phase independently to each other. After the qualification phase, only qualified teams selected by the organizers (see Section 2.5) 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 30 k€, divided in several categories as explained at `https://www.roadef.org/challenge/2022/en/prix.php`

## 2.3 Datasets

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

- Dataset A of 30 instances at the beginning of the challenge.
- Dataset B of 40 instances available after the end of the sprint phase phase.
- Dataset C of 50 instances available after the end of the qualification phase.

- Dataset X of 100 hidden instances that will be used to rank the candidates for the final phase and available after the end of the challenge.

Datasets A and B contains small/medium size instances while datasets C and X contain regular size instances.

## 2.4 Evaluation

At the deadline of the sprint phase, teams must submit their solution files. The dataset A will be used to compare teams.

At the deadline of the qualification phase, each team must send its computer program (see Section 2.6), solution files and a short document (2 pages max), which describes the solving method, the characteristics of the computer used to find solutions and the computational results table. The dataset B will be used to compare teams.

At the deadline of the final phase, each team must send its computer program (see Section 2.6), solution files and a short document (5 pages max), which describes the solving method, the characteristics of the computer used to find solutions and the computational results table. The datasets C and X will be used to compare teams.

The computer program of each team will be executed on each instance with the time limit documented in the instance's parameters file.

## 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 instance, depending on their result. We denote by score the number of points earned by a team, whereas result denotes 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 objective value of team $i$ for instance $j$. Let $nb_i^j$ be the number of teams with objective value strictly better than the objective value of team $i$ on instance $j$ :
$nb_i^j = \text{Card}(\{k \neq i \text{ s.t. } y_{kj} < y_{ij}\})$.

Let $R$ be the maximal score that a team can earn form one instance.$R$ will be set at the start of each phase, according to the number of participants. Let $p_{ij}$ be the number of points scored (*i.e.* the score) by team $i$ for the instance $j$. It is defined by

$$p_{ij} = \begin{cases} \text{max(0, R - nb}_i^j) & \text{if the solution is feasible,} \\ 0 & \text{if the solution is unfeasible,} \\ -1 & \text{if a crash occurs.} \end{cases} \tag{1}$$

The global score of a team, denoted as $score(i)$ is defined by :
$score(i) = \sum_{j \in [1,m]} p_{ij}$

The winner is the team with the highest score. In case of equality (i.e. there are several teams with the same highest score), the winner is the team with the highest number of best known solutions.

### 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 | unfeasible | 29 | 41 | 55 | 10 | 5 |
| Team 7 | 39 | 30 | 43 | crash | 10 | 4 |

Table 1: Values of $y_{ij}$

The values $nb_i^j$ 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 $nb_i^j$

The scores $p_{ij}$ 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 | 5 | 4 | 3 |
| Team 7 | 0 | 3 | 2 | -1 | 4 | 5 |

Table 3: Values of $p_{ij}$

Finally, the global scores are:

| | |
|---|---|
| Team 1 | 15 |
| Team 2 | 11 |
| Team 3 | 18 |
| Team 4 | 7 |
| Team 5 | 20 |
| Team 6 | 20 |
| Team 7 | 13 |

Table 4: Values of $score(i)$

In this example, Team 5 would be the winner, it has the same score than Team 6 (20) but a higher number of best known solutions (2 vs 0).

### 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. It must return an output solution file using standards defined in Section 3.2. 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.

Programs of candidates will be run on Google Cloud Platform, in VM with 8 CPU, 32 GB of RAM and CentOS 7. The following list provides the solvers allowed during the challenge:

- CPLEX
- GUROBI 9.5 and later
- CPO
- LocalSolver

# 3 Files format

The files format refer to the notations defined in the subject of the challenge.

## 3.1 Input files

An instance (one instance per plant) contains 3 files :

- The items to be delivered from suppliers to a plant.
- The planned trucks from suppliers to the plant.
- The parameters for the optimization.

### 3.1.1 Items file

It is a csv file and it contains a headers line with the titles of the columns. The package code is used only for Renault's analysis. All the items that can be packed together in the same stack share identical maximal stackability, maximal density and maximal weight on the bottom item.

| Field | Type | Example | Comments |
|---|---|---|---|
| Item ident | char | 00900160_20221201_2314 | |
| Supplier code ($IU_i$) | char | 0007768904 | |
| Supplier dock ($IK_i$) | char | 171C12 | |
| Plant code ($IP_i$) | char | 0090017100 | |
| Plant dock ($IG_i$) | char | 171C36A | |
| Product code ($IR_i$) | char | 963027763R | |
| Package code | char | SFDA–6670 | |
| Number of items | int | 20 | |
| Length ($IL_i$) | int | 1200 | mm |
| Width ($IW_i$) | int | 1000 | mm |
| Height ($IH_i$) | int | 800 | mm |
| Weight ($IM_i$) | float | 120,45 | Kg |
| Nesting height ($\widehat{IH}_i$) | int | 10 | mm |
| Stackability code ($IS_i$) | char | SFDA—6670-1200-1000 | |
| Forced orientation ($IO_i$) | char | lengthwise/widthwise/none | |
| Earliest arrival time ($IDE_i$) | int | 202109230000 | 23/09/2021 00:00 |
| Latest arrival time ($IDL_i$) | int | 202109271315 | 27/09/2021 13:15 |
| Inventory cost ($IC_i$) | int | 6 | |
| Max stackability ($ISM_i$) | int | 5 | |

### 3.1.2 Planned trucks file

It is a csv file and it contains a headers line with the titles of the columns. Planned trucks are defined for every n-uplet supplier/plant/product/arrival time. Characteristics of a truck (arrival time, plant code, dimensions, max weights, max density, stack with multiple docks, distances) depend only on the truck id. In other words, for the same truck id, all these characteristics are identical, whatever the supplier, the docks or the product.

| Field | Type | Example | Comments |
|---|---|---|---|
| Code of supplier $u$ ($u \in \widetilde{TU}_t$) | char | 0007768904 | |
| Supplier loading order ($\widetilde{TE_t}$) | int | 2 | |
| Supplier dock $k$ ($k \in \widetilde{TK_{ut}}$) | char | 171C12 | |
| Supplier dock loading order ($TKE_{ut}$) | int | 1 | |
| Plant code ($TP_t$) | char | 0090017100 | |
| Plant dock $g$ ($g \in \widetilde{TG_{pt}}$) | char | 171C36A | |
| Plant dock loading order ($TGE_{pt}$) | int | 3 | |
| Product code ($\in \widetilde{TR_t}$) | char | 963027763R | |
| Arrival time ($TDA_t$) | int | 202109271315 | |
| Id truck | char | P380411201 | |
| Length ($TL_t$) | int | 13400 | mm |
| Width ($TW_t$) | int | 2444 | mm |
| Height ($TH_t$) | int | 3100 | mm |
| Max weight ($TM_t^m$) | int | 24000 | kg |
| Stack with multiple docks ($TF_t$) | int | 1=yes / 0=no | |
| Max stack density ($TEM_t$) | int | 1500 | Kg/m$^2$ |
| Max weight on the bottom item in stacks ($TMM_t$) | int | 750 | Kg |
| Cost ($TC_t$) | int | 1500 | EUR |
| Middle axle max weight ($EM^{mm}$) | int | 12000 | kg |
| Rear axle max weight ($EM^{mr}$) | int | 31500 | Kg |
| weight of the tractor ($CM$) | float | 7808 | Kg |
| Distance between front and middle axles ($CJ^{fm}$) | int | 3800 | mm |
| Distance between front axle and center of gravity of the tractor ($CJ^{fc}$) | int | 1040 | mm |
| Distance between front axle and harness of the tractor ($CJ^{fh}$) | int | 3330 | mm |
| Weight of the empty trailer ($EM$) | int | 7300 | Kg |
| Distance between harness and rear axle of the trailer ($EJ^{hr}$) | int | 7630 | mm |
| Distance between trailer's center of gravity and rear axle ($EJ^{cr}$) | int | 2350 | mm |
| Distance between end of trailer and harness ($EJ^{eh}$) | int | 1670 | mm |

#### 3.1.3 Parameters file

It is a csv file with as single line and it contains a headers line with the titles of the columns.

| Field | Type | Example | Comments |
|---|---|---|---|
| **Cost coefficients** | | | |
| Coefficient inventory cost ($\alpha^I$) | float | 10,2 | |
| Coefficient transportation cost ($\alpha^T$) | float | 1,0 | |
| Coefficient cost extra truck ($\alpha^E$) | float | 0,2 | |
| **Other parameters** | | | |
| Runtime limit | int | 3600 | Seconds |

### 3.2 Output files

#### 3.2.1 Trucks file

It is a csv file and it contains the headers line "Id truck;Loaded length;Weight of loaded items;Volume of loaded items;EMm;EMr" (titles of the columns). The file contains planned and extra trucks which are used to deliver the items. The truck id is the truck primary key.

| Field | Type | Example | Comments |
|---|---|---|---|
| Id truck | char | P380411201 | |
| Loaded length | int | 12500 mm | $max_{s \in \widetilde{TS}_t} sx_s^e$ |
| Weight of loaded items | float | 1894,31 | kg |
| Volume of loaded items | float | 14,544 | $m^3$ |
| Weight on the middle axle of the trailer ($em^m$) | float | 1875,83 | kg |
| Weight on the rear axle of the trailer ($em^r$) | float | 18,47 | kg |

The ident of an extra truck is based on the ident of its associated planned truck. For instance, the second extra truck of a planned truck with id=P380411201, will have the id Q380411201_2.

### 3.2.2 Stacks file

It is a csv file and it contains the headers line "Id truck;Id stack;Stack code;X origin;Y origin;Z origin;X extremity;Y extremity;Z extremity" (titles of the columns). The file contains all the stacks which are loaded into trucks. The stack id is the stack primary key.

| Field | Type | Example | Comments |
|---|---|---|---|
| Id truck | char | P380411201 | Must be defined in the output trucks file |
| Id stack | char | P380411201_1 | |
| Stack code | char | A | To be used for display (cf FIGURE 1) |
| X origin ($sx_s^o$) | int | 0 | mm |
| Y origin ($sy_s^o$) | int | 0 | mm |
| Z origin ($sz_s^o$) | int | 0 | mm |
| X extremity ($sx_s^e$) | int | 1010 | mm |
| Y extremity ($sy_s^e$) | int | 1206 | mm |
| Z extremity ($sz_s^e$) | int | 407 | mm |

The stack id is based on the ident of the truck into which it is loaded. The stack code ranges from A to Z, then from AA to AZ, then from BA to BZ etc. The rule to assign stack code to stacks : ascending X origin $sx_s^o$, then ascending Y origin $sy_s^o$.

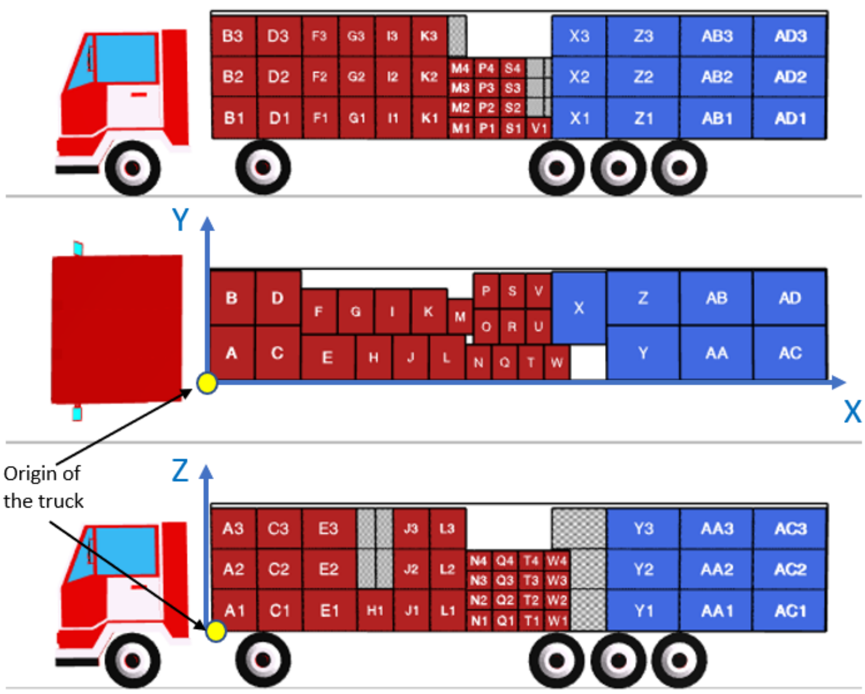


Figure 1: Display of stack codes and item codes : there are 96 items (codes : A1 to AD3) and 30 stacks (codes : A to AD).

### 3.2.3 Items file

It is a csv file and it contains the headers line "Id truck;Id stack;Stack code;X origin;Y origin;Z origin;X extremity;Y extremity;Z extremity" (titles of the columns). The file defines the truck and the stack into which an item is packed. The item primary key is represented by the n-uplet supplier/plant/product/earliest/latest arrival time/id truck/id stack/item code.

| Field | Type | Example | Comments |
|---|---|---|---|
| Item ident | char | 00900160_20221201_2314 | Must be defined in the input items file |
| Id truck | char | P380411201 | Must be defined in the output trucks file |
| Id stack | char | P380411201_1 | Must be defined in the output stacks file |
| Item code | char | A1 | To be used for display (cf FIGURE 1) |
| X origin | int | 0 | mm |
| Y origin | int | 0 | mm |
| Z origin | int | 0 | mm |
| X extremity | int | 1010 | mm |
| Y extremity | int | 1206 | mm |
| Z extremity | int | 407 | mm |

### 3.3 Checker

A checker is provided. It is written in java and is available in github at https://github.com/renault-iaa/challenge-roadef-2022.git. The jar file includes the source code.

### 3.4 Visualizer

A visualizer of trucks loading plan is provided in github at https://github.com/renault-iaa/challenge-roadef-2022.git. It is written in C and C#, and runs only on windows platforms.

## 4 Discussion and updates

To help participants managing their project, the organizers try to communicate immediately changes that might occur during the challenge by updating the dedicated github repository. 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. Renault 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 off the challenge scope without the author team's agreement.

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

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