Challenge ROADEF' 2005
Car Sequencing Problem

Alain NGUYEN

RENAULT - alain.nguyen@renault.com

1-PROBLEM DESCRIPTION 3
1.1-CONTEXT 3
1.1.1 – Planning and scheduling process 3
1.1.2 –The paint shop requirements 3
1.1.3 –The assembly line requirements 3
1.1.4 – Role of production day D-1 4
1.2-PROBLEM TO SOLVE 4
1.2.1 – Multi-objective optimization with « top priority level to ratio constraints » 5
1.2.2 – Multi-objective optimization with « top priority level to paint color batches » 5
1.2.3 – Details on the computation of the violations number of ratio constraints 5
1.2.4 –"Easy-to-satisfy" and "difficult-to-satisfy" high priority level ratio constraints 7
2 – HOW TO PARTICIPATE ? 8
2.1 – SCENARIOS TEST SETS PROVIDED BY RENAULT 8
2.1.1 – Scenarios Test Sets 8
2.1.2 – Content of a Test Set 8
2.1.2 – File formats 8
2.1.2.1 File of the limitation on the upper batch size of paint color batches 9
2.1.2.2 The optimization objectives file 9
2.1.2.3 The ratio constraints file 10
2.1.2.4 The vehicles file 11
2.2 – INFORMATION TO BE PROVIDED BY THE CANDIDATES 12
2.2.1-Documentation 12
2.2.2 – Solutions files 13
2.2.3 - Programs 13
2.2.4 – Organization of a candidate's file hierarchy 13
2.3 – THE EVALUATION AND RANKING PROCEDURE 14
2.3.1 – Limit runtime and test environments 14
2.3.2 – The evaluation program 14
2.3.3 – Evaluation method of a scenario 14
2.3.3.1 – A scenario with « top priority level to ratio constraints » 15
2.3.3.2 – A scenario with « top priority level to paint color batches » 15
2.3.4 – Global evaluation of a scenarios Test Set 16
2.4 – TIMETABLE 16

1-Problem description

1.1-Context

1.1.1 – Planning and scheduling process

Customer orders are transmitted daily in real-time to car factories.

The daily job of factories is (1) to assign a production day to each ordered vehicle, according to production line capacities and delivery dates which were promised to customers by salesmen. Factories have then (2) to schedule the order of cars to be put on the line for each production day, while satisfying as many requirements as possible of the plant shops: body shop, paint shop and assembly line.

We will focus on the requirements of the paint shop and the assembly line, since the body shop does not set requirements for the daily schedule.

While scheduling each production day, the set of vehicles of the production day, which is determined at stage (1), cannot be changed.

An industrial application manages the planning/scheduling process, using Linear Programming for stage (1) and Simulated Annealing for stage (2).

1.1.2 –The paint shop requirements

The paint shop has to minimize the consumption of paint solvent. The paint solvent is used to wash spray guns each time the paint color is changed between 2 consecutive scheduled vehicles.

Therefore there is a requirement to group vehicles together by paint color, that is to minimize the number of paint color changes in the sequence of scheduled vehicles. In other words, we try to minimize spray guns washes, i.e. to schedule the longest paint color batches possible.

Paint color batches have a limitation on the upper batch size, since spray guns have to be washed regularly, even though there are no paint color changes in the sequence of scheduled vehicles. This limitation is a **hard constraint**.

1.1.3 –The assembly line requirements

In order to smooth the workload on the assembly line, vehicles that require special assembly operations have to be evenly distributed throughout the total processed cars. These vehicles are

considered to be "hard to assemble". These cars may not exceed a given quota over any sequence of vehicles.

This requirement is modeled by a **ratio constraint N/P**. Ratio constraints are associated with car characteristics which require extra operations on the assembly line (for instance, sunroof, air conditioning etc).

A ratio constraint N/P means that at most N cars in each consecutive sequence of P are associated with the constraint. For instance, if N/P = 3/5, there must not be more than 3 constrained cars on any consecutive sequence of 5 vehicles.

If N/P=1/P, it means that 2 constrained cars must be separated by at least P-1 consecutive non-constrained vehicles. For instance, with a ratio constraint 1/5: X _ _ _ _ X is an acceptable sequence, where 'X' is a vehicle associated with the ratio constraint and '_' is a car not concerned by the ratio constraint.

There are two classes of ratio constraints, high priority level constraints and low priority level constraints. High priority level ratio constraints are due to car characteristics that require a heavy workload on the assembly line. Low priority level ratio constraints result from car characteristics that cause small inconvenience to production. High priority level ratio constraints must be satisfied preferentially to low priority level constraints.

Ratio constraints are **soft constraints**: the full respect of all the ratio constraints cannot be ensured beforehand when a production day is scheduled. The problem may be over-constrained. Hence the optimization objective is to minimize the violations number of ratio constraints.

1.1.4 – Role of production day D-1

When production day D is scheduled, the ultimate scheduled vehicles of production day D-1 must be taken into account. Vehicles of production day D-1 are already scheduled. Therefore their positions cannot be changed when production day D is scheduled. The computation of the violations number of ratio constraints on production day D must take into account the ultimate scheduled vehicles of production day D-1. Production day D+1 is ignored while production day D is scheduled.

1.2-Problem to solve

We have to schedule a sequence of vehicles that satisfies best the paint shop and the assembly line requirements. 2 strategies can be followed depending on car factories:

- « **top priority level to ratio constraints** »: the assembly line requirements have a higher priority level than those of the paint shop
- « **top priority level to paint color batches** »: the paint shop requirements are more critical than those of the assembly line.

1.2.1 – Multi-objective optimization with « top priority level to ratio constraints »

We have to optimize the following objectives, from the objective with the highest priority level to the objective with the lowest priority level, and **without any compensation** between objectives, in order to schedule vehicles of a production day:

1. to minimize the violations number of high priority level ratio constraints
2. to minimize the number of paint color changes
3. to minimize the violations number of low priority level ratio constraints

or

1. **to minimize the violations number of high priority level ratio constraints**
2. **to minimize the violations number of low priority level ratio constraints**
3. **to minimize the number of paint color changes**

or

1. **to minimize the violations number of high priority level ratio constraints**
2. **to minimize the number of paint color changes**

Some factories do not declare low priority level ratio constraints.

1.2.2 – Multi-objective optimization with « top priority level to paint color batches »

We have to optimize the following objectives, from the highest priority level to the lowest, and **without any compensation** between objectives, in order to schedule vehicles of a production day:

(1) To minimize the number of paint color changes
(2) To minimize the violations number of high priority level ratio constraints
(3) To minimize the violations number of low priority level ratio constraints.

or

(1) To minimize the number of paint color changes
(2) To minimize the violations number of high priority level ratio constraints

For the 2 multi-objective optimizations, there is only one hard constraint: the limitation on the upper batch size of paint color batches.

There must not be any compensation between objectives: the optimization of the objective with rank O+1 must not lower the result of the objective optimization with rank O.

1.2.3 – Details on the computation of the violations number of ratio constraints

The best solution for a ratio constraint N/P would be a scheduled production day where there are at most N constrained vehicles on any consecutive sequence of P vehicles. The best solution for a ratio constraint 1/P would be a scheduled production day where any 2 constrained cars are separated by at least P-1 non-constrained vehicles: for instance, X _ _ _

X is an acceptable sequence for a ratio constraint 1/4, where X is a car associated with the ratio constraint and '_' is a vehicle not concerned by the ratio constraint.

If we cannot generate a schedule without any violation of a ratio constraint, constrained vehicles must be spaced out as much as possible throughout the total scheduled cars. The purpose is to smooth the workload on the workstations of the assembly line.

We try to obtain this "space out" distribution of constrained vehicles by computing violations of ratio constraints on rolling sequences of the production day. For a sequence of constrained vehicles, the lesser these vehicles are spaced out, the higher the violations number will be. Because there are violations in many more rolling sequences, and the violations number may be higher in each rolling sequence.

Example: ratio constraint 1/5 => the violations number of the ratio constraint is computed on sequences of 5 vehicles
_ _ _ X _ _ _ X : 1 violation on the sequence _ _ _ X _ _ _ X
_ _ _ X _ _X _: total of 2 violations on the rolling sequences _ _ _ X _ _X _ and _ _ _ X _ _X _

Number of violations of a ratio constraint on a sequence = (number of vehicles associated with the ratio constraint on the sequence) – (ratio constraint numerator) **if (number of vehicles associated with the ratio constraint on the sequence) > (ratio constraint numerator)** **= 0 otherwise**

This computation of the violations number applies for ratio constraints 1/P and ratio constraints N/P (with N<>1).

Exemple: ratio constraint 1/5
_ _ X X _ _ X _: there are 2 violations on the sequence: (3 constrained vehicles) – 1 (= ratio constraint numerator)

The optimization objective is to minimize the total number of violations of ratio constraints over all the rolling sequences of the production day.

While scheduling production day D, the ultimate scheduled vehicles of production day D-1 must be taken into account for the computation of the violations number of ratio constraints on production day D. It implies that rolling sequences (on which the violations number of ratio constraints is computed) must begin on production day D-1. For a ratio constraint N/P, the first rolling sequence contains the P-1 ultimate scheduled vehicles of production day D-1 and the first scheduled vehicle of production day D. It must be remembered that vehicles of production day D-1 are already scheduled, so their positions cannot be changed.

Example: the first rolling sequence for a ratio constraint 1/5
XXXXXY_ _ _ _
X: ultimate scheduled vehicles of production day D-1
Y: first scheduled car of production day D

Since production day D+1 is ignored while scheduling production day D, for a ratio constraint N/P, the ultimate rolling sequence of production day D contains the P ultimate scheduled cars of production day D. This ultimate rolling sequence begins at position L-P, with L as the number of vehicles of production day D and with positions ranging from 0 to L-1.

Production day D+1 is ignored while scheduling production day D, but for each ratio constraint, we want to compute the violations occurred with the ultimate cars of production day D, as if the first scheduled vehicles of production day D+1 are not associated with the ratio constraint.

For a ratio constraint N/P, the ultimate rolling sequences will have a length comprised between P-1 and (N+1). If in one these rolling sequences, the number of cars associated with the ratio constraint is strictly greater than N, then there will be violations, whether the first scheduled vehicles of production day D+1 are associated or not with the ratio constraint.

This computation method enables to give the same weight to the violations occurred with the ultimate P-1 cars of production day D (for a ratio constraint N/P) as to the violations on the previous vehicles.

Example: the ultimate rolling sequences for a ratio constraint 1/5
YYYYYYYY
YYYYYYYY
YYYYYYYY
Y: ultimate scheduled vehicles of production day D

1.2.4 –"Easy-to-satisfy" and "difficult-to-satisfy" high priority level ratio constraints

RENAULT will provide scenarios with high priority level ratio constraints, which are « easy-to-satisfy» or « difficult-to-satisfy ». This will have an impact on the evaluation of the candidates' solutions. The classification « easy-to-satisfy » vs. « difficult-to-satisfy » does not apply to low priority level ratio constraints.

A scenario is considered to have **easy-to-satisfy** high priority level ratio constraints if the RENAULT industrial application generates a schedule of the production day without any violation of ratio constraints. Easy-to-satisfy high priority level ratio constraints are feasible.

A scenario is considered to have **difficult-to-satisfy** high priority level ratio constraints if the RENAULT industrial application cannot generate a schedule of the production day without any violation of high priority level ratio constraints.
It occurs when the number of constrained vehicles overwhelms ratio constraints: for instance, the production day contains 25% of constrained vehicles and there is a high priority level ratio constraint 1/5.
Generally speaking, the difficult-to-satisfy high priority level ratio constraints can be feasible or not.

2 – How to participate?

2.1 – Scenarios Test Sets provided by RENAULT

2.1.1 – Scenarios Test Sets

3 scenarios Test Sets will be provided by RENAULT. They will give opportunities to the candidates to fine-tune their programs:

- Test Set A: available since the beginning of the challenge. The jury will use these data to select the finalists. This is the **qualification Test Set**.
- Test Set B: will be provided to the finalists to fine-tune their programs for the Test Set X.
- Test Set X: will be used to rank the finalists at the ROADEF'2005 conference. This is the **final Test Set**. The Test Set X will stay unknown to the finalists until the end of the challenge.

The fine-tuning of programs should improve the robustness of the algorithms from data variability. This robustness is critical for industrial scheduling applications. It is unmanageable for IS teams to fine-tune programs whenever there are major changes in production data (ratio constraints, vehicles characteristics etc) in any RENAULT factory.

2.1.2 – Content of a Test Set

A set of scenarios is provided; each scenario represents production data from a factory. The same factory may be used to produce several scenarios.

A scenario contains:

- High priority level ratio constraints
- Low priority level ratio constraints
- The limitation on the upper batch size of paint color batches
- Vehicles of production day D and the ultimate scheduled cars of production day D-1 (there are DenMax-1 vehicles of production day D-1, with DenMax as the maximum of denominators of ratio constraints).
- A vehicle is defined by: an identifier, the date of production day (date D or date D-1), the paint color and a flag (0/1) for each ratio constraint, which indicates whether the vehicle is associated or not with the ratio constraint, the sequence rank scheduled by the industrial application from RENAULT.
- The ranking of optimization objectives.

Some scenarios may not contain any low priority level ratio constraint. Some factories may give a high priority level to all their ratio constraints.

2.1.2 – File formats

A scenario is a directory named instance_XXXXX containing the following files:

- File of the limitation on the upper batch size of paint color batches: « paint_batch_limit.txt »
- The optimization objectives file: « optimization_objectives.txt »
- The ratio constraints file: « ratios.txt »
- The vehicles file: « vehicles.txt «

They are ascii text files with semi colon (';') as delimiter.

2.1.2.1 File of the limitation on the upper batch size of paint color batches

This file contains the limitation on the upper batch size of paint color batches.

Example of the content of the file « *paint_batch_limit.txt* »:
limitation;
10;

2.1.2.2 The optimization objectives file

The file contains the ranking of optimization objectives, from the objective with the highest priority level to the objective with the lowest priority level. The file contains a list of values « rank ; objective name ». The objective name indicates whether high priority level ratio constraints are easy-to-satisfy or difficult-to-satisfy.

Several examples of the content of the file « optimization_objectives.txt »:

File content in case of « top priority level to ratio constraints » and easy-to-satisfy high priority level ratio constraints

rank;objective name;
1;high_priority_level_and_easy_to_satisfy_ratio_constraints;
2;paint_color_batches;
3;low_priority_level_ratio_constraints;

or

rank;objective name;
1;high_priority_level_and_easy_to_satisfy_ratio_constraints;
2;low_priority_level_ratio_constraints;
3;paint_color_batches;

or

rank;objective name;
1;high_priority_level_and_easy_to_satisfy_ratio_constraints;
2;paint_color_batches;

File content in case of « top priority level to paint color batches » and easy-to-satisfy high priority level ratio constraints

rank;objective name;
1;paint_color_batches;
2;high_priority_level_and_easy_to_satisfy_ratio_constraints;
3;low_priority_level_ratio_constraints;

or

rank;objective name;
1;paint_color_batches;
2;high_priority_level_and_easy_to_satisfy_ratio_constraints;

File content in case of « top priority level to ratio constraints » and difficult-to-satisfy high priority level ratio constraints

rank;objective name;
1;high_priority_level_and_difficult_to_satisfy_ratio_constraints;
2;paint_color_batches;
3;low_priority_level_ratio_constraints;

or

rank;objective name;
1 ;high_priority_level_and_difficult_to_satisfy_ratio_constraints;
2;paint_color_batches;

or

1;high_priority_level_and_difficult_to_satisfy_ratio_constraints;
2;low_priority_level_ratio_constraints;
3;paint_color_batches;

2.1.2.3 The ratio constraints file

The file contains high priority level and low priority level ratio constraints. Each ratio constraint is defined by a ratio N/P, a priority level flag (1=high priority level, 0=low priority level) and an identifier (the rank of the ratio constraint).

Example of the content of the file « ratios.txt »:

Ratio Prio
; ; Ident;
1/2; 1; HPRC1;
1/8; 1; HPRC2;
2/3; 1; HPRC3;
1/4; 1; HPRC4;
3/4; 0; LPRC1;
1/3; 0; LPRC2;
2/3; 0; LPRC3;

In this example, there are 4 high priority level ratio constraints and 3 low priority level ratio constraints.

2.1.2.4 The vehicles file

The file contains vehicles of production day D and the ultimate scheduled cars of production day D-1.

For each vehicle, there are the following fields:

1. **Date:** production date of the vehicle, with the format "AAAA WW D", where AAAA is the year, WW is the industrial week in the year (WW=1..52), and D is the day in the week (1 for Monday, 2 for Tuesday, etc.)
2. **SeqRank**: sequence rank: the rank of the vehicle in the production day. This rank was scheduled by RENAULT industrial application. Ranks in a production day begin at number 1.
3. **Ident**: identifier of the vehicle.
4. **Paint Color**: paint color code of the vehicle.
5. **HPRCi**: '1' if the vehicle is associated with the high priority level ratio constraint HPRCi, '0' otherwise. Number of columns HPRCi = number of high priority level ratio constraints.
6. **LPRCi**: the same with low priority level ratio constraints.

Only the ultimate scheduled vehicles of production day D-1 are provided. Only these ultimate cars are needed to compute the violations number of ratio constraints in production day D. The number of vehicles of production day D-1 = (the maximum of denominators of ratio constraints) - 1.

Example of the content of the file « vehicles.txt »:

Date;SeqRank;Ident;Paint Color;HPRC1; HPRC2; HPRC3;HPRC4;LPRC1;LPRC2; LPRC3
2003 18 3;997;025031830079;5;1;0;0;0;0;0;0
2003 18 3;998;025031850339;2;0;0;0;0;1;0;0
2003 18 3;999;025031920802;1;1;0;1;1;1;1;1
2003 18 3;1000;025031830269;1;0;0;1;0;1;0
2003 18 3;1001;025031850094;2;0;0;0;0;0;0
2003 18 3;1002;025031910010;2;1;1;0;0;1;0
2003 18 3;1003;025031850332;2;0;0;0;0;1;0
2003 18 3;1004;025031820317;1;1;0;1;1;1;1
2003 18 5;1;025031910825;3;0;0;1;0;1;0;1;0
2003 18 5;2;025031820293;7;1;0;1;1;1;0;1;0
2003 18 5;3;025031850497;2;0;0;0;0;0;0;0;0
2003 18 5;4;025031911085;5;1;0;1;0;1;1;1;0
2003 18 5;5;025031850299;5;0;0;1;0;1;0;1;0
2003 18 5;6;025031850082;2;1;0;0;0;0;0;0;0
2003 18 5;7;025031850783;5;0;0;1;0;1;1;1;0
2003 18 5;8;025031920415;5;1;1;1;1;0;0;1;0
2003 18 5;9;025032010270;5;0;0;0;0;0;0;0;0
2003 18 5;10;025031911083;5;1;0;1;0;1;1;1
2003 18 5;11;025031910060;5;0;0;0;0;0;0;1

2003 18 5;12;025031970126;5;1;0;1;1;0;0;1
2003 18 5;13;025031910182;2;0;0;0;0;1;0;0
2003 18 5;14;025031850845;1;1;0;1;0;1;1;1
2003 18 5;15;025031910373;1;0;0;1;0;0;0;1
2003 18 5;16;025031850597;1;1;0;0;0;1;0;0
2003 18 5;17;025031830304;1;0;0;1;1;1;0;1
2003 18 5;18;025031850802;1;1;0;1;0;1;1;1
2003 18 5;19;025031850666;2;0;0;0;0;0;0;0
2003 18 5;20;025031850746;2;1;0;0;0;1;0;0
2003 18 5;21;025031850856;2;0;0;0;0;1;0;0
2003 18 5;22;025031830202;9;1;0;1;1;1;0;1
2003 18 5;23;025031850767;2;0;0;0;0;0;0;0

In the example above, there are 4 high priority level ratio constraints, 3 low priority level ratio constraints, 8 vehicles of production day D-1 and 23 cars of production day D.

Date of production day D-1: « *2003 18 3* »
Date of production day D: « *2003 18 5* »

As demonstrated in the example, dates of production days D and D-1 are not always consecutive. They are working dates in factories, without Sundays and holidays. In the example, the date « 2003 18 4 » is May 1st 2003 which is a holiday.

For each vehicle, fields of high priority level ratio constraints are defined before fields of low priority level ratio constraints. The sequence of ratio constraints fields is identical to the sequence of ratio constraints in the file "ratios.txt".

Vehicles of production day D-1 are defined before cars of production day D.

2.2 – Information to be provided by the candidates

2.2.1-Documentation

- Description of the team members:
  - name(s),
  - position(s) (student, post doc, professor, private sector employee, ...),
  - affiliation,
- brief description of the team and their past experience;
- description of the methods used with motivation and references (4-6 pages), including the commercial software and computer languages used;
- a synthesis (an excel table) of the best solutions obtained within the given time, **i.e. 600 seconds CPU time** (on a machine similar to the PC described thereafter). This excel table must have the following fields:

Scenario name	Number of violations of high priority level ratio constraints	Number of violations of low priority level ratio constraints	Number of paint color changes	Evaluation of the scenario

The evaluation of the scenario will be computed according to the method presented thereafter.

For non-deterministic programs, 10 runs per scenario are required and the candidates will give the median solutions.

2.2.2 – Solutions files

Solutions files corresponding to different scenarios must be provided. A solution file must have the following fields: « Sequence rank; Identifier ». Sequence ranks begin at 1. The file solution will have the same name as the scenario.

Example of the content of a solution file:

```
 1;025031910825
 2;025031820293
 3;025031850497
 4;025031911085
 5;025031850299
 6;025031850082
 7;025031850783
 8;025031920415
 9;025032010270
10;025031911083
11;025031910060
12;025031970126
```

2.2.3 - Programs

The candidates will provide programs, either:

- an executable code which could be run directly on the machine provided by RENAULT, or
- the C or C++ code including a manual for compiling, and the associated *makefile* allowing RENAULT to create an executable code.

2.2.4 – Organization of a candidate's file hierarchy

For a candidate indexed `NN`, the file hierarchy is:

- main directory: `Candidate-NN/`
- sub-directories:

```
Candidate-NN/Team-description/
Candidate-NN/Method-description/
Candidate-NN/Result-synthesis/
Candidate-NN/Instances/
Candidate-NN/Solutions/
```

```
Candidate-NN/Results/
Candidate-NN/Program/
```

Whether furnished or generated, the executable must be usable online by typing in the directory `Candidate-NN/Program/`:

```
executable-name instance-name -t cpu-time
```

with:

`instance-name`: scenario name.
`cpu-time`: **run time allocated in seconds, limited to 600.**

The program should look after input data of scenario `instance-name` in the directory `Candidate-NN/Instances/instance-name` and write the solution obtained at `cpu-time` in the directory `Candidate-NN/Solutions/`. The solution file will be named `instance-name`.

2.3 – The evaluation and ranking procedure

There are 2 categories:

- **Senior** category: all the candidates belong to this category,
- **Junior** category: limited student projects selected by the jury (a project is a student project if the majority of the team members are students, possibly advised by their professors).

Remark

Note that a student project could win the prize of the senior category if this project is the best over all the submitted projects. On the opposite, a senior project cannot win the prize of the junior category.

2.3.1 – Limit runtime and test environments

Runtime will be limited to **600 seconds** on a **PC Pentium4/1.6 Ghz/1 Go RAM**. This limitation represents the maximal response time accepted by industrial users.

Programs can run on either Win2000 or Linux Redhat 7.3.

Commercial software with license keys are forbidden.

2.3.2 – The evaluation program

The evaluation program from RENAULT, for each scenario, will call the program of a candidate, let it run **600 seconds**, look for the solution file obtained in the directory `Candidate-NN/Solutions/`, check and evaluate it, and write the evaluation of the solution in the directory `Candidate-NN/Results/`.

For non-deterministic programs, 10 runs per scenario are required to obtain the median solution.

2.3.3 – Evaluation method of a scenario

We present thereafter the evaluation method for the qualification stage. There may be minor changes of the evaluation method for the final stage.

A mark will be computed for each scenario, weighting the different optimization objectives of the scenario.

2.3.3.1 – A scenario with « top priority level to ratio constraints »

Ranking of optimization objectives:
1. to minimize the violations number of high priority level ratio constraints
2. to minimize the number of paint color changes
3. to minimize the violations number of low priority level ratio constraints

mark of the scenario
= (10000*number_of_violations_of_high_priority_level_ratio_constraints)
+ (100*number_of_paint_color_changes)
+ (number_of_violations_of_low_priority_level_ratio_constraints)

or

1. to minimize the violations number of high priority level ratio constraints
2. to minimize the violations number of low priority level ratio constraints
3. to minimize the number of paint color changes

mark of the scenario
= (10000*number_of_violations_of_high_priority_level_ratio_constraints)
+ (100* number_of_violations_of_low_priority_level_ratio_constraints)
+ (number_of_paint_color_changes)

or

1. to minimize the violations number of high priority level ratio constraints
2. to minimize the number of paint color changes

mark of the scenario
= (10000*number_of_violations_of_high_priority_level_ratio_constraints)
+ (100*number_of_paint_color_changes)

2.3.3.2 – A scenario with « top priority level to paint color batches »

Ranking of optimization objectives:
(1) to minimize the number of paint color changes
(2) to minimize the violations number of high priority level ratio constraints
(3) to minimize the violations number of low priority level ratio constraints

mark of the scenario
= (10000* number_of_paint_color_changes)
+ (100* number_of_violations_of_high_priority_level_ratio_constraints)
+ (number_of_violations_of_low_priority_level_ratio_constraints)

or

(1) to minimize the number of paint color changes
(2) to minimize the violations number of high priority level ratio constraints

mark of the scenario
= (10000* number_of_paint_color_changes)
+ (100* number_of_violations_of_high_priority_level_ratio_constraints)

2.3.4 – Global evaluation of a scenarios Test Set

For each Test Set, there are 3 classes of scenarios:

- scenarios with « top priority level to ratio constraints » and easy-to-satisfy high priority level ratio constraints
- scenarios with « top priority level to ratio constraints » and difficult-to-satisfy high priority level ratio constraints
- scenarios with « top priority level to paint color batches »

For each class of scenarios, a **median** of the marks obtained on all the scenarios belonging to the same class is evaluated (the mark of a scenario is computed according to the evaluation method presented at 2.3.3).

Over the 3 classes, a global mark is computed to rank the candidates:

$$\sum_{class} \left[\frac{(candidate_mark(class) - worst_mark(class))}{(best_mark(class) - worst_mark(class))} \right]$$

The « worst mark » and the « best mark » are the worst of all the marks obtained by the candidates, and the best of all the marks obtained by the candidates, for a given class of scenarios.

This global mark is the only criteria to rank the candidates. This evaluation method should erase any numerical gap between the median marks of the 3 classes of scenarios.

Based on the results on Test Set A, a maximum **10 candidates** will be selected for the final stage.

Candidates of the final stage will be ranked according to the results on Test Set X. The Test Set B is only provided for the fine-tuning of the programs by the candidates.

2.4 – Timetable

15/07/2003: beginning of the qualification stage
problem description available on the WEB and **1 scenario provided per class** (i.e. a total of 3 scenarios)

01/10/2003: all scenarios of Test Set A available on the WEB

01/12/2003: application deadline of the candidates,
please send your complete affiliations to Van-Dat CUNG

29/02/2004: **end of the qualification stage**
deadline for candidates to submit their results and programs on Test Set A

17/05/2004 : **beginning of the final stage**
- qualification stage results announced, selection of the finalists who will be invited to present their works at the ROADEF'2005 conference
- Test Set B available on the WEB

This Test Set B is provided to the finalists for fine-tuning their programs.

03/10/2004: deadline for the candidates to send an upgraded release of their programs
RENAULT will test the candidates' programs on Test Set X (which is unknown to the candidates)

ROADEF'2005 conference: end of the finale stage
Announcement of the final results on Test Set X.
Candidates present their works.
The winner in each of the 2 categories will receive a prize sponsored by the ROADEF.