AUTOMAIN

Augmented Usage of Track by Optimisation of Maintenance, Allocation and Inspection of Railway Networks

Project Reference: 265722
Research Area: SST.2010.5.2-1

Tool in operation for demonstration of results WP 3&4

Deliverable Reference No: D5.2

Due Date of Deliverable: 31/01/2014
Actual Submission Date: 27/01/2014

Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Document Summary Sheet

This document presents the planning and scheduling tool developed within WP5. After a brief introduction in Section I, Section II presents the tool and its components. Section III illustrates different use cases (demonstrations), and Section IV discusses how results from WP3 and WP4 can be assessed by the planning and scheduling tool. Finally, Section V provides some conclusions on the results achieved, the potential improvements and future perspectives.

Additional information	
Subproject / workpackage:	**WP5**
Author:	**Damian Diaz (TU-BS)** **Federico Grasso Toro (TU-BS)** **Henri Olink (ProRail)** **François Ramond (SNCF)** **Raja Rebaï (EURODECISION)** **Renaud Rivoallan (EURODECISION)** **Clive Roberts (UoB)**
Document date:	**27/01/2014**
Status:	**Final, Approved by consortium**

History			
V0.1	**11/12/2013**	**first draft**	**Initial draft version**
V0.2	**18/12/2013**	**2nd draft**	**Second draft version**
V0.3	**19/12/2013**	**3rd draft**	**Third draft version**
V0.4	**20/12/2013**	**4th draft**	**Fourth draft version**
V1.0	**17/01/2014**	**5th draft**	**Submitted to PMT and WP5 members**
V1.1	**27/01/2014**	**final**	**Final version**

Table of Contents

Executive Summary

This report describes the planning and scheduling tool developed within WP5 of AUTOMAIN in order to automatically generate optimized planning and scheduling solutions.

Finding good planning solutions and good schedules is not an easy task because the number of potential solutions is extremely high, due to the combinatorial nature of the underlying optimization problem. For this reason, basic planning approaches are usually limited and do not enable to obtain the best results. Enumerating and evaluating all possible solutions was also not considered as a feasible option because it would have been very inefficient in terms of computation time. Given these elements, it was decided to study how dedicated optimization approaches, based on operations research techniques, could be used to generate optimized planning and scheduling solutions.

Consequently, the early stages of the work performed in WP5 were dedicated to specifying the expectations from such a tool, and to designing specific optimization algorithms. An original problem decomposition was proposed, with synchronization mechanisms. This was described in deliverable D5.1.

Then, these algorithms were implemented into software source code to build a planning and scheduling tool. The current report focuses on the uses of the tool and on some results obtained during test phases.

I. Introduction

a) *Importance of effective planning and scheduling in regard of the possession time*

As for any heavy industrial work, track maintenance is an activity that requires a strong organization in order to make sure that all the work which is expected to be done can actually be performed, on time. Given the fact that railways security depends on track condition, planning maintenance operations is particularly important, at least to make sure that all operations can be performed before their deadline, considering all production and logistics aspects.

Beyond this necessity, planning in an effective way is also required to make sure that the possession time is not higher than it could be. Indeed, minimizing the possession time enables to maximize the throughput of the network, and hence the traffic and the associated revenue generated by the traffic. In this respect, it should be noted that, with equivalent possession times, different maintenance schedules can lead to different impacts on the commercial traffic, so this aspect should be considered as well, in order to further minimize the impact of maintenance on commercial traffic.

b) *General presentation of the work performed in the Work Package*

WP5 was dedicated to the development of innovative algorithms in order to automatically generate planning and scheduling solutions to reduce the possession time required for track maintenance. To be tested and demonstrated, these algorithms had to be embedded within software including many other auxiliary aspects and functionalities, such as a business model, input data parsing, output data and diagrams generation and communication between layers. Different types of work were, therefore, involved in WP5: modelling, software design, input/output definition, source code writing, testing and debugging, etc. Moreover, due to a lack of available data when the tool was beginning to run, WP5 also had to generate a first dataset combining all sorts of information (see Section II-b), in an XML format specifically tailored for the needs of a track maintenance planning and scheduling tool. This initial dataset was inspired by data provided by SNCF. It was used throughout the project to help with refining and calibrating the algorithms.

II. Tool presentation

a) Overview

The planning and scheduling tool is a command line tool: it should be started from a computer terminal, using its different options to provide input data and set some parameters. Once the command is issued, the tool runs for a certain time, providing some information on the screen, and finally generates output files describing the results obtained after computation.

b) Data input / output

As represented on Figure 1, the tool requires different types of information (input):

- Network description
    - Lines, tracks with their speed, switches
- Trains expected to run on this network
    - Route and timing of each train
- Maintenance operations to be performed
    - Type of activity, location, deadline
- Machines available to perform the maintenance operations
    - Number, type of activity, speed, unavailability periods (if any)

These input data are read by the tool in the early stages of each run, and are used to precisely describe the type of planning solutions to be generated.

The output results are composed of a global planning of all operations, describing for each their starting/ending dates, the assigned machine, all machine travels between successive operations, and a detailed schedule of every operation, with its impact on the commercial traffic (cancelled or delayed trains, if any).

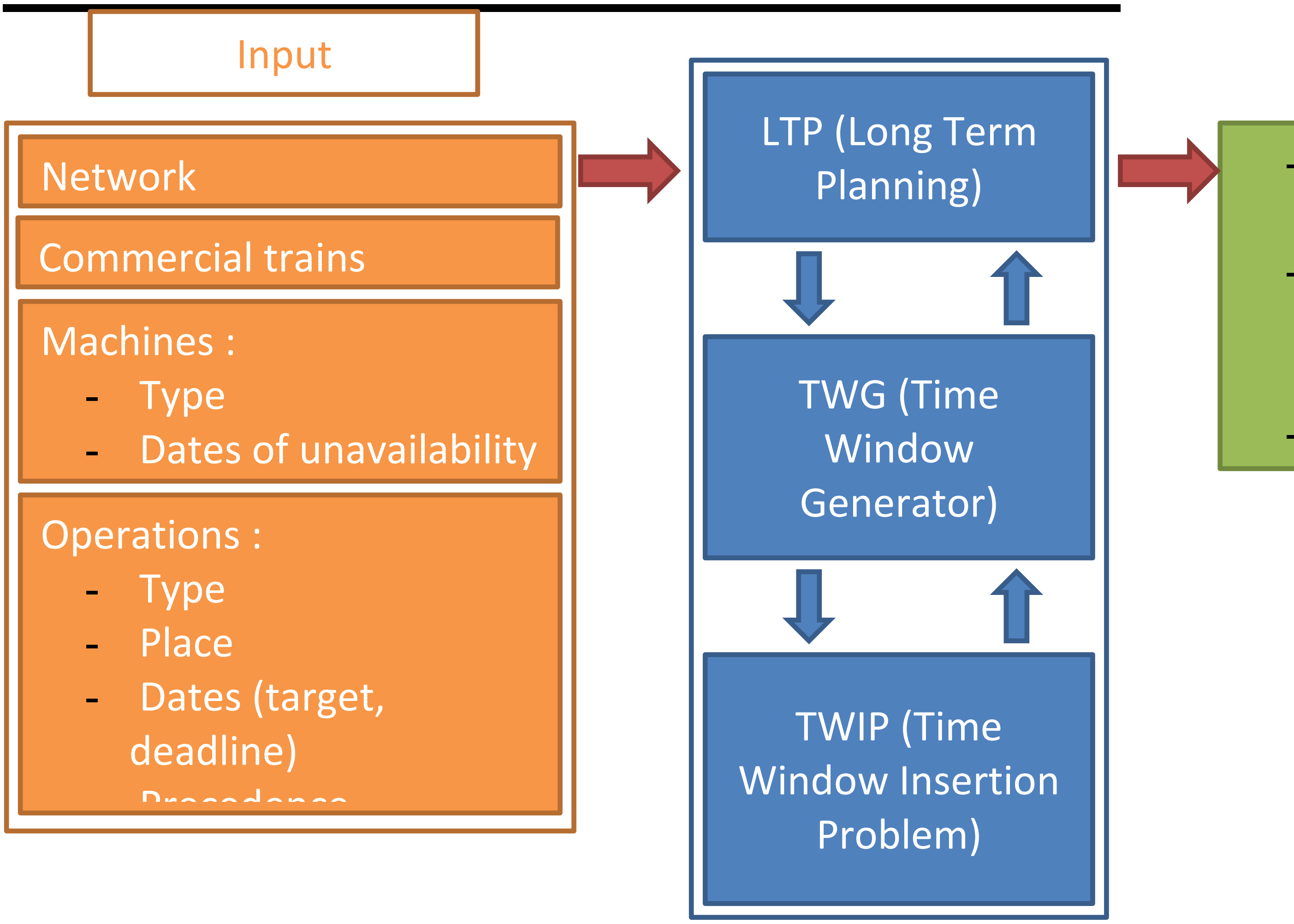


Figure 1 - Input/output

c) *Architecture*

The tool developed by WP5 relies on three main modules, each performing a specific task: LTP (Long-Term Planning), TWG (Time-Window Generator) and TWIP (Time-Window Insertion Problem). These modules work in cooperation with each other, to be able to handle different time horizons simultaneously. More precisely, the LTP module calls TWG to validate its assumptions, and TWG in turn uses TWIP to finely insert a short possession into an existing timetable.

These three modules are described more in-depth in the next sections.

d) *Long-Term Planning (LTP) module overview*

Objectives

The objectives to be optimised by LTP are a combination of objectives related to:

1. possession time (minimise number of days of possession required to perform all operations);
2. machine travel (minimise number of kilometres travelled between operations);
3. impact on commercial trains (if any, minimise delays and cancelations);
4. maintenance dates (perform maintenance at a date as close as possible to its given target).

Out of these four objectives, objective 1 is the only one strictly derived from the global AUTOMAIN objective, which is to reduce possession time. Indeed, we believe that a model with objective 1 only would lead to unrealistic solutions being produced by the tool because it might lead to planning solutions which have a heavy impact on traffic (train delays and/or cancellations), which are very inefficient from the maintenance point of view or which require unnecessary transport of machines. The role of objectives 2 to 4 is to avoid such poor solutions being proposed by the algorithms.

Given the computation time constraint, we adopt a specific three-step approach to solve the LTP problem. The idea behind this decomposition is to design, in a short computation time, a first good solution by implicitly approximating microscopic constraints. After this first "draft" solution, TWG and TWIP are called either to confirm the choices that were estimated (by refining the details of the schedule) or to reject some of the assumptions which were made during the estimation. Once this is done for all operations, planning changes are tried iteratively to improve the solution until the allocated computation time is over.

Scope

The LTP problem aims at proposing long-term planning solutions by deciding, for each operation to be performed, a machine, a start date and an end date. It also needs to ensure that these macroscopic planning solutions respect constraints at more microscopic levels, which are local constraints over 24h periods, explicitly handled by the other modules, TWG and TWIP.

One of the greatest challenges for LTP is the lack of fixed data concerning the duration of both operations and routings of the machine. Indeed, all these durations are only valid for a certain configuration associated with the assigned machine, the potential beginning dates of the operation, the simultaneity of other operations, the expected commercial traffic, etc. Every time one of these parameters changes, the durations must be recomputed through calls to TWG and, in turn, TWIP, since only these latter modules are precise enough to provide reliable information on the durations.

Traditional optimisation approaches are based on the assumption that all required data are available as soon as the algorithms start. In the case of LTP, due to the reasons stated above, this assumption does not hold. Moreover, the computation effort induced by a call to TWG (and subsequently to TWIP) requires significant computation time. Consequently, the number of calls to TWG must be minimised and the cases where TWG is called must be chosen carefully among all possibilities.

Inputs

The data used as input of LTP are composed of:

- network description
  - Tracks, nodes between tracks, with their respective maximal allowed speeds
- maintenance resources (machines able to perform maintenance operations)
  - Type of maintenance
  - Travel speed
  - Work speed
- maintenance operations
  - Type of work to be done
  - Localisation of each operation
  - Ideal (target) date to start the operation
  - Deadline (latest date when the operation must be started)
  - Priority level
  - Potentially, predecessors : references of maintenance operations which must be performed previously (for instance, measurement must be done at least 6 weeks prior to tamping)

Even though information regarding commercial trains is expected in the input file, LTP does not use them directly: this aspect is handled by the two other modules (TWG and TWIP).

Outputs

The output of LTP is an assignment of dates to each maintenance operation (start date & end date) along with a plan (Gantt diagram) of the use of available maintenance machines over the time horizon.

The individual maintenance machine plan must be feasible in terms of maintenance operations (the machines should be able to perform the operations between their assigned start and end dates) and in terms of routing (the machines should have sufficient time to be moved over the railroad network between two successive maintenance operations). Operations must be performed before their deadline, and, obviously, cannot be performed by machines during their own maintenance. The precedence constraints, if any, must be respected. Moreover, two operations cannot be executed simultaneously if they are located on overlapping sections of the same track.

e) Time Window Generation (TWG) module overview

Objectives

The TWG module aims to connect the LTP module and the TWIP module. The main tasks accomplished by the TWG module are the maintenance operation splitting and the optimal maintenance machine path calculation.

On the one hand the LTP module deals with the long-term planning of the railroad system maintenance on a macroscopic level. Several maintenance operations must be performed along several months and the LTP module requires a method to fine-schedule maintenance operations.

On the other hand the TWIP module deals with the fine-scheduling and validation of one short maintenance operation on a microscopic level. This operation must be completed in less than 24 hours and the maintenance machine path used for the operation must be predefined.

The time-window granted by the LTP module to schedule one maintenance operation is usually several days long. Therefore the TWG module adapts this medium-term scheduling problem, converting the requested task into small short-term problems for the TWIP module to solve. Also the TWG module calculates the necessary maintenance machine paths.

Scope

TWG handles the medium-term scheduling problem provided by LTP, dealing with the transformation of the medium-term scheduling problem into short-term scheduling problems. It also handles the subsequent solving of these short-term problems by TWIP.

The main functions of the TWG module are:

- Splitting one maintenance operation into several short maintenance operations.
- Finding a suitable depot to store the maintenance machine between short maintenance operations.
- Calculating the optimal path of the maintenance machine between the storage depot and the maintenance operation location.
- Generating short time-windows to schedule each short maintenance operation.

The TWG module uses simple heuristic rules in order to find an adequate splitting of the maintenance operation.

To achieve the splitting of the maintenance operation, two different strategies are used:

- The main splitting strategy consists in dividing the maintenance workload evenly along each day of the medium-term scheduling time-window.

- The secondary splitting strategy consists in assigning as much maintenance work as possible along each day of the medium-term scheduling time-window, without cancelling trains.

The main splitting strategy is always used. The secondary splitting strategy is only used when no solution was found after applying the main splitting strategy.
This combination of strategies is designed to achieve additional solutions when the cancellation of commercial trains is not allowed.

The short maintenance operations are subject to duration constraints. Short maintenance operations longer than 2 hours and shorter than 8 hours are preferred in the current version of the TWG module.

The splitting strategies take into account the duration constraints of the short maintenance operations.

The storage depot used to store the maintenance machine between short operations should be located near to the maintenance operation location in order to reduce the travel time.

A suitable storage depot is found by performing a *Breadth First Search* for the depot located nearest to the maintenance operation location.

Optimal paths must be found in order to minimize the travel time of the maintenance machine. The calculation of the optimal route for the maintenance machine is performed using a modified version of *Dijkstra's algorithm*.

Different maximal speeds for forward and reverse travelling directions are taken into consideration during the optimal route calculation. This leads to a preference for paths that follow the forward direction.

Finally the TWG module generates a time-window shorter than 24 hours to use the TWIP module to validate and fine-schedule each short maintenance operation.

Inputs

The inputs of the TWG module are:

- Reference data, including:
  - Railway network definition.
  - Characteristics of all maintenance machines.
  - Characteristics of all maintenance operations.
- Assigned maintenance machine data, including:
  - Travelling and working speed.
  - Type of machine.
  - Initial location.
- Maintenance operation data, including:

  - Type of operation.
  - Location of the operation.
  - Length of the operation.
- Scheduling time-windows data (medium-term):
  - Initial date (Machine available date).
  - Desired end date (Target end date).
  - Latest allowed end date (Latest end date).

The railway network definition constrains the possible paths that the maintenance machine can follow. The calculated optimal paths are based on the shape of the railway network.

There are two medium-term time-windows that the TWG module takes into consideration. One medium-term time-window is defined by the interval *[Machine available date, Target end date]*. The other medium-term time window is defined by the interval *[Machine available date, Latest end date]*.

Outputs

The TWG module outputs two solutions, one for each considered medium-term time-window.

Each solution is composed of:

- A group of fine-scheduled short maintenance operations, that together conform the whole maintenance operation.
- The path followed by the maintenance machine during the complete maintenance operation, starting at the machine's initial location and ending at the storage depot.
- A list of modifications performed by TWIP on the commercial trains' schedules in order to schedule the short maintenance operations (cancelled and delayed trains).

f) TWIP (Time-window insertion)

Objectives

The time window insertion problem (TWIP) aims to find the best way to insert a maintenance path on a given day between the commercial train paths.

The departure and arrival dates of the maintenance train at each point have to be computed by TWIP. The most important objective of TWIP is to minimize the possession time of a maintenance machine. The possession time corresponds to the time that the maintenance train takes to go from a depot to the maintenance location, do the effective work, and return to the end depot.

To insert a maintenance train, TWIP can delay commercial trains, or even cancel some of them.

If the cancellation of a commercial train is undesirable, an option can be activated in order to forbid the cancellation of commercial trains. In this case TWIP may not find a feasible solution.

The best solution is the one which minimises both the possession time and the impact of the modification on the commercial timetable.

Scopes

TWIP is a short-term scheduling problem for which the temporal horizon is one day (the optimisation day). The temporal horizon can also be a subset of this day, if the time period of the day when the operations must be placed is known.

TWIP cannot modify the maintenance train path. This path is computed by TWG. TWIP has to set the maintenance train schedule.

TWIP is only able to manage maintenance tasks from a given point A to a given point B (A≠B) and is not able to treat punctual maintenance (at a specific location only).

Inputs

TWIP is based on shared common data “Reference Data” with some additional data provided by TWG (maintenance train and task).

The Reference Data contains a list of commercial trains with their path and schedule. The commercial path is fixed and cannot be changed by the solver. The schedule can be modified (the train can be delayed or even cancelled).

TWG provides 2 elements which are the maintenance short operation that has to be optimized and the train that will be used in order to perform the maintenance short operation.

A Boolean allows the “no impact on commercial traffic” mode of TWIP to be chosen.

Outputs

The results of TWIP are the time windows for the maintenance train and the maintenance task.
If TWIP has to modify (delay or cancel) the commercial timetable, TWIP returns to TWG the list of delayed trains associated with the new schedule, and the list of cancelled trains.

Constraints

There are three main classes of constraints:

- For commercial trains
    - The time deviation between the originally scheduled time of a train and its rescheduled time must remain below a given value which depends on the type of the train. Otherwise, the train must be cancelled.

    - The speed of a train is bounded by both a minimal and a maximal speed. Both values depend on the type of the train.
    - Trains must slow down if there is a maintenance task on an adjacent track.
- For maintenance machines
    - Each move of a maintenance machine has an earliest start time and a latest completion time.
    - The speed of the maintenance train is limited. This value depends on the type of maintenance train.
- For maintenance tasks
    - The task must be performed between a given earliest start time and a given latest completion time.
    - Some tasks must be done before others (precedence constraint).
    - Some tasks must be performed using a maintenance machine (resource constraint).
    - Some tasks must be performed simultaneously.

The definition of TWIP also adds constraints of non-conflict between different elements (maintenance tasks, commercial trains or moves of maintenance machines) that occur on the tracks. There must always be a safety time between the time-windows when they share the same infrastructure element. This time depends on the type of the time-windows.

Objective function

<u>Normal mode</u>

The objective function takes into account several criteria with specific costs.

The cancelCost (cost to cancel a commercial train) and the variation cost (the cost of 1 second of delay of a commercial train) are defined in the xml input file (arguments of object TrainPath)

The makespan cost is proportional to the cancelCost.
One hour of makespan is equivalent to one commercial train cancelation.

<u>« No impact on commercial traffic » mode</u>

In this mode, the priority is to ensure that no trains are cancelled.
The cancelling cost of a commercial train is very high in this mode (10000x higher) to avoid the possibility of compensating for the train cancellation with a gain in possession time.

III. Demonstration over test cases

a) Long-term planning optimization

This test case corresponds to a call to the LTP module only, without any detailed local validation by TWG and TWIP. It generates a long-term planning where possession time is optimized.

The tool is launched with the following command line:

```
$ ./automain LTP data/demo.xml data/refdataxml.xsd 20 noValidation
```

Where

- ./automain is the name of the tool (executable file),
- LTP the module called,
- data/demo.xml the input data file,
- data/refdataxml.xsd the xml schema document file,
- 20 the time, in minutes, allowed to LTP (optimization runs for 20 min before being stopped),
- And noValidation is a flag used to skip validation of operations by TWG and TWIP.

Once launched, the tool starts processing the data and running the algorithms:

```
19/12/13 16:56:41 INFO AUTOMAIN CALLED WITH THE FOLLOWING COMMAND LINE:
19/12/13 16:56:41 INFO ./automain LTP data/demo.xml data/refdataxml.xsd 20 noValidation
19/12/13 16:56:41 INFO ************************************
19/12/13 16:56:41 INFO data/demo.xml validates data/refdataxml.xsd
19/12/13 16:56:41 INFO Reading data/demo.xml...
19/12/13 16:56:54 INFO Nb depot = 721
19/12/13 16:56:54 INFO Nb location = 8400
19/12/13 16:56:54 INFO Nb point = 704
19/12/13 16:56:54 INFO Nb track = 1274
19/12/13 16:56:54 INFO Nb trackGroup = 788
19/12/13 16:56:54 INFO Nb train = 277
19/12/13 16:56:54 INFO Mean nb etp per train = 53.0505
19/12/13 16:56:54 INFO Nb maintenance op = 200
19/12/13 16:56:54 INFO Total length of tracks (m) = 6693878
19/12/13 16:56:54 INFO File data/demo.xml was read.
19/12/13 16:56:54 INFO ************************************
19/12/13 16:56:54 INFO Graph is connex : explored 704 points.
19/12/13 16:56:54 INFO Calling LTP...
19/12/13 16:56:54   DEBUG ###### Predecessors ######
19/12/13 16:56:54   DEBUG Ope Measurement_before_TampingOp_0 predecessor of Ope TampingOp_0
19/12/13 16:56:54   DEBUG Ope Measurement_before_TampingOp_1 predecessor of Ope TampingOp_1
19/12/13 16:56:54   DEBUG Ope Measurement_before_TampingOp_10 predecessor of Ope TampingOp_10
19/12/13 16:56:54   DEBUG Ope Measurement_before_TampingOp_100 predecessor of Ope TampingOp_100
[...]
19/12/13 16:56:54   DEBUG Ope TampingOp_77 and Ope TampingOp_96 aren't compatible.
19/12/13 16:56:54   DEBUG Ope TampingOp_81 and Ope TampingOp_86 aren't compatible.
19/12/13 16:56:54   DEBUG Ope TampingOp_83 and Ope TampingOp_93 aren't compatible.
19/12/13 16:56:54   DEBUG ####################################
19/12/13 16:56:54  ALGO Starting planning build...
19/12/13 16:56:54  ALGO Planning built ok.
19/12/13 16:56:54  ALGO Planning constraints OK.
19/12/13 16:56:54  ALGO Planning cost : 495006
19/12/13 16:56:54  ALGO Possession cost : 442000
19/12/13 16:56:54  ALGO Lower bound for possession cost : 217000
19/12/13 16:56:54  ALGO Starting simulated annealing...
19/12/13 17:16:55  ALGO Simulated annealing ok.
19/12/13 17:16:55  ALGO Planning constraints OK.
19/12/13 17:16:55  ALGO Planning cost : 439724
19/12/13 17:16:55  ALGO Possession cost : 405000
19/12/13 17:16:55 INFO Planning exported to out/ltp.csv
19/12/13 17:16:55 INFO LTP was called.
```

During 20 minutes, the possession time is reduced by iteratively making changes in the planning that result in performing more operations during the same possessions (same day, same track on non-overlapping sections).

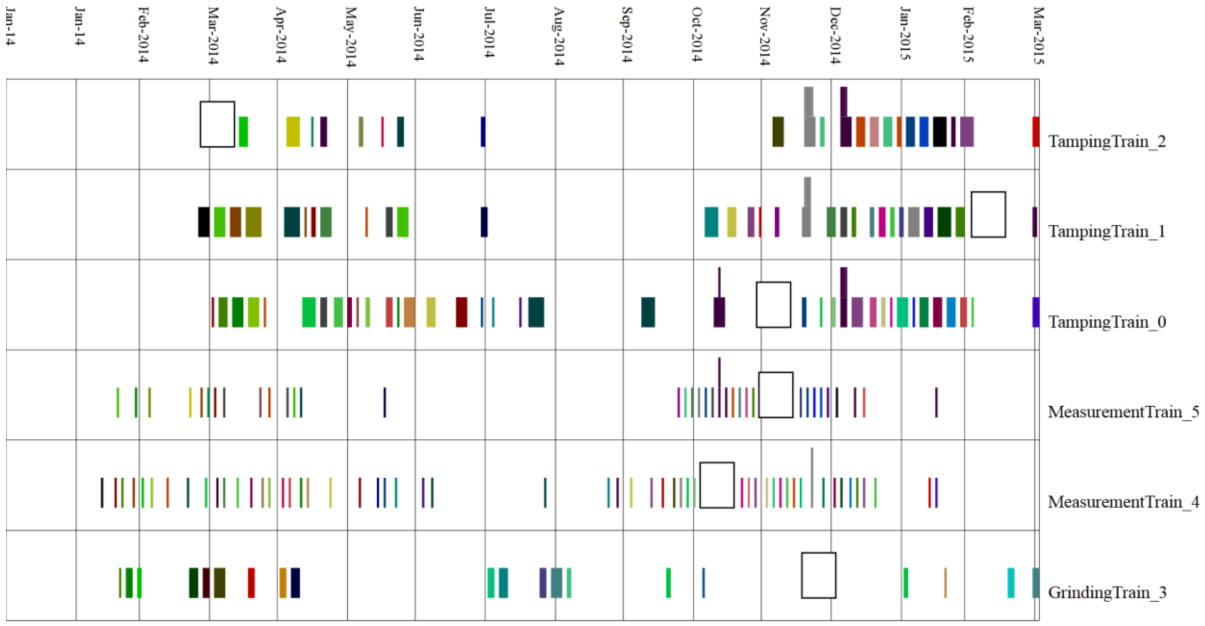


Figure 2 – Long-term planning before optimization

In this example, the possession time required to do all 200 operations is 442 days. This initial planning is obtained by applying basic planning rules, without using any optimization technique. The associated Gantt chart is represented on Figure XXX. Each coloured rectangle represents an operation performed by a maintenance train (Y-axis) over a certain period (X-axis), while empty rectangles represent periods where machines are unavailable due to their own maintenance. Note that, for a given maintenance train, a minimum time is respected between two consecutive operations. We can see that a few operations are combined with each other: these combinations are represented on the chart by higher rectangles on the concerned days. This aspect is not explicitly optimized but it is also not avoided, so it occurs naturally for a few operations in this example.

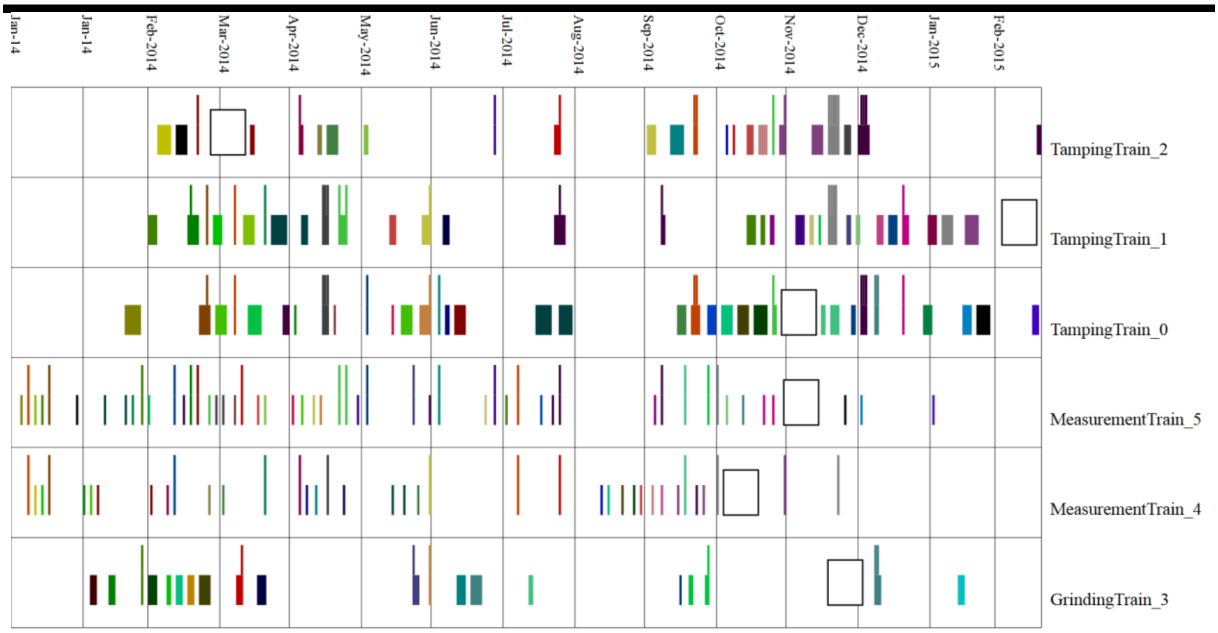


Figure 3 - Long-term planning obtained after optimization

After 20 min of computation, the generated planning contains more combined operations (higher rectangles), meaning that it requires less possession time. On this example, the optimized planning requires 405 days of possession, to be compared with 442 days before optimization, resulting in a gain of 8% of possession time. If we let the algorithm run longer, we reach a planning requiring 402 days of possession (9% gain) after 10 hours, this value being reached after 37 minutes and no further improvement occurring later.

b) Full mode: LTP optimization and validation of each operation

The full mode corresponds to a call to the three modules (LTP, TWG and TWIP). This test was performed with the demo example.

We can summarize the demo data set characteristics by the following:

- Number of depots : 721
- Number of locations = 8400
- Number of points = 704
- Number of tracks = 1274
- Number of track Groups = 788
- Number of trains = 277
- Mean number of elementary train paths per train = 53.0505
- Number of maintenance operation to insert = 200
- Total length of tracks (m) = 6693878

This test was run on a computer with the following properties:

- 4 core at 3 GHz
- 16 GB of RAM
- Windows 7 64 bits

The solver used by TWIP is IBM Ilog CPLEX v12.5.1

The overall loop takes almost 54 hours of computation:

- Start time : 20/12/13 10:17:18
- End time : 22/12/13 16:11:52

During this optimization TWIP was called 722 times.

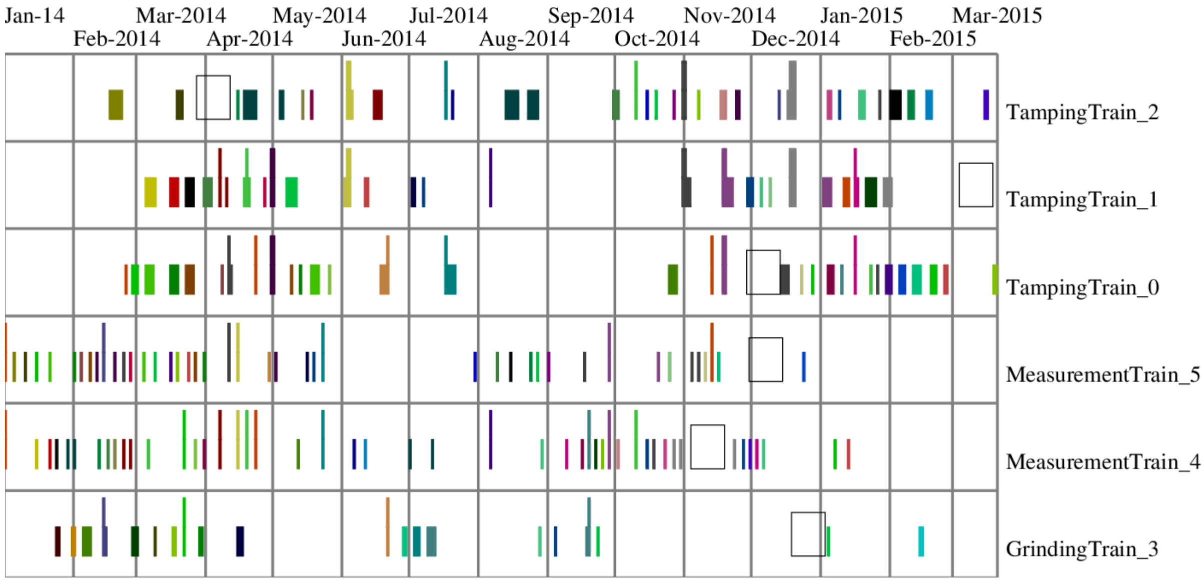


Figure 4 - Long-term planning obtained after optimization and validation of each operation

Figure 3 above represents the planning obtained after optimization and validation of each operation, with calls to TWG / TWIP.

c) *Validation of an operation – example 1*

This test case corresponds to the first validation call performed by the LTP module.
The maintenance operation provided by the LTP module is fine-scheduled and validated in a joint effort of the TWG and TWIP modules.

The tool is launched with the following command line:

```
./automain LTP data/demo.xml data/refdata.xsd 5
```

In this case it was possible to validate the maintenance operation without modifying the commercial schedules. The example presents a situation where the maintenance operation has short duration and there is low commercial traffic around the maintenance operation location.

The validation process starts with a validation call from the LTP module to the TWG module.

The inputs received by the TWG module from the LTP module are:

- Reference data
- Assigned maintenance machine data, including:
    - Travel speed: 100 km/h
    - Work speed: 8 km/h
    - Initial location: at the beginning of Tr_628
- Maintenance operation data, including:
    - Start location of the operation: at 17355m from the beginning of Tr_608
    - End location of the operation: at 5136m from the beginning of Tr_608
    - Length of the operation: 12219m
- Scheduling time-window (medium-term):
    - Machine available date: 10/01/2014, at 0:00 hours
    - Target end date: 13/01/2014, at 0:00 hours
    - Latest end date: 13/01/2014, at 0:00 hours

In this example the *Latest end date* and the *Target end date* are equal; therefore the two solutions calculated by the TWG module are the same.

The estimated duration of the maintenance operation is 1 hour 32 minutes. This duration is shorter than the minimum short operation duration constraint therefore only one short maintenance operation is created.

A search for a storage depot is performed. The depot nearest to the location of the maintenance operation is found at the beginning of Tr_605.

The optimal path is calculated from the initial location of the maintenance machine to the maintenance operation location and from the maintenance operation location to the storage depot.

A small extract of the tool text output describing the path is presented below:

```
Path(1) Name: Path starting at 0@Tr_628, doing work between 17355@Tr_608 and 5136@Tr_608,
ending at 0@Tr_605.
 -> 0 meters from track Tr_628
 -> 5570 meters from track Tr_628
 -> 6300 meters from track Tr_632
 -> 6100 meters from track Tr_634
 -> 9900 meters from track Tr_636
 ...
 -> 3636 meters from track Tr_598
 -> 16378 meters from track Tr_614
 -> 21754 meters from track Tr_616
 -> 17355 meters from track Tr_608
 -> 5136 meters from track Tr_608
 -> 0 meters from track Tr_605
```

A time-window shorter than 24 hours is generated, limiting the scheduling of the short maintenance operation to:

- Minimum Start Time: 10/01/2014, at 0:00 hours
- Maximum End Time: 10/01/2014, at 23:59 hours

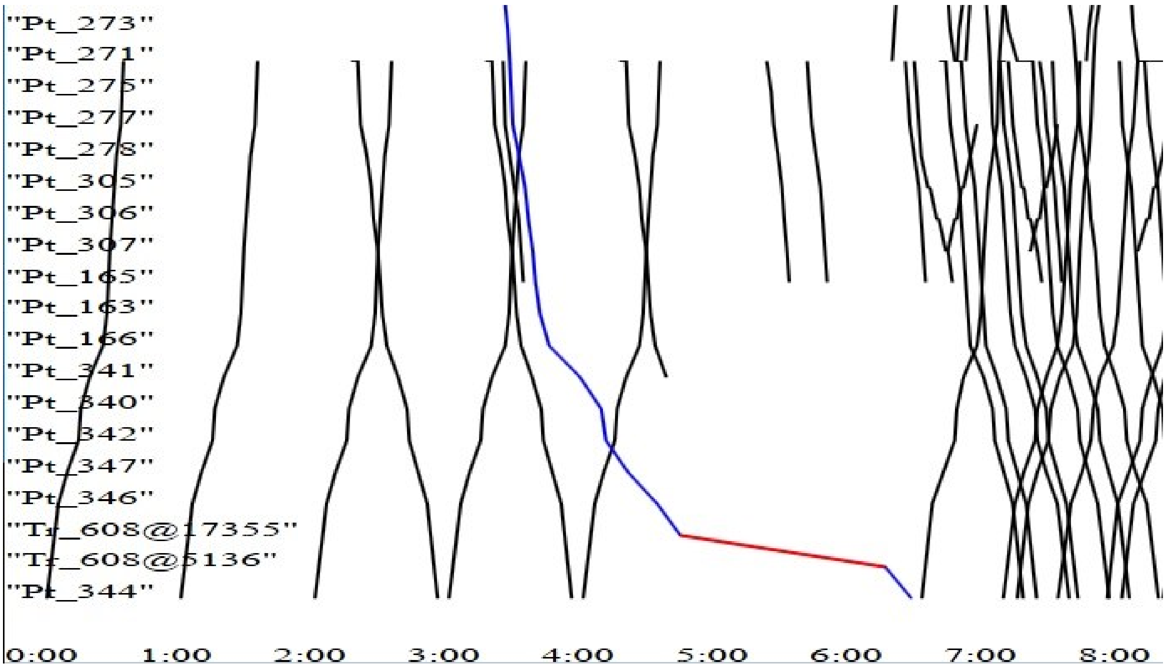
Finally, the TWIP module is used to fine-schedule the short maintenance operation.

TWIP Input Data

The maintenance short operation is “Measurement_before_TampingOp_40”. It consists of a Measurement to do between locations 17355 and 5136 on track “Tr_608”. TWIP has to insert this operation on January 10th, 2014.

The maintenance train associated with this operation is “MeasurementTrain_5” (Travel Speed: 100 km/h; Work Speed: 8 km/h)

TWIP Graphical Output



TWIP Numerical results

- 0 train is cancelled
- 0 train is delayed
- The maintenance train starts at 02:21:11 and ends at 06:45:46 (makespan : 04:24:35)
- The maintenance operation starts at 05:02:19 and ends at 06:33:58 (duration : 01:31:39)
- The solution’s cost equals 32605

TWIP Results Analysis

TWIP is able to insert the maintenance without cancelling or delaying commercial trains.
TWIP benefits from the “empty space” in the commercial timetable between 4 AM and 6:30 AM.

From points Pt_271 to point Pt_165, we can see that TWIP is able to respect headway between the maintenance Train and the previous commercial train which is travelling on the same path.

On the time-space diagram, the maintenance train crosses 2 commercial trains that travel in the opposite direction. These commercial trains actually travel on parallel tracks. There is no conflict between the maintenance train and these trains.

For this example the TWIP module reported that the short maintenance operation scheduling was feasible and that no commercial train was cancelled.

In this case the validation process finalizes after fine-scheduling this short maintenance operation.

Composition of the TWG module output

The TWG module composes its output based on splitting of the maintenance operation, the calculated maintenance machine path and the scheduling results from the TWIP module.

For this example the output of the TWG module is composed of:

- A list of cancelled trains: No cancelled trains
- A list of delayed trains: No delayed trains
- One short maintenance operation with the following schedule:
    - Operation Start: 10/01/2014, at 05:02:19
    - Operation End: 10/01/2014, at 06:33:58
- The full maintenance train path with a defined schedule.
- Objective value: 32605

This output is the value returned to the LTP module, when the TWG module is called by the LTP module.

d) Validation of an operation – example 2

This test case corresponds to the second validation call performed by the LTP module.
The maintenance operation provided by the LTP module is fine-scheduled and validated in a joint effort of the TWG and TWIP modules.

The tool is launched with the following command line:

```
./automain LTP data/demo.xml data/refdata.xsd 5
```

In this case one commercial train was delayed in order to schedule the maintenance operation. The example presents a situation where a previously scheduled maintenance operation has to be taken into account.

The validation process starts with a validation call from the LTP module to the TWG module.

The inputs received by the TWG module from the LTP module are:

- Reference data
- Assigned maintenance machine data, including:
    - Travel speed: 100 km/h
    - Work speed: 8 km/h
    - Initial location: at the beginning of Tr_628
- Maintenance operation data, including:
    - Start location of the operation: at 5193 m from the beginning of Tr_134
    - End location of the operation: at 15250 m from the beginning of Tr_134
    - Length of the operation: 10057 m
- Scheduling time-window (medium-term):
    - Machine available date: 10/01/2014, at 0:00 hours
    - Target end date: 13/01/2014, at 0:00 hours
    - Latest end date: 13/01/2014, at 0:00 hours

In this example the *Latest end date* and the *Target end date* are equal; therefore the two solutions calculated by the TWG module are the same.

The estimated duration of the maintenance operation is 1 hour 15 minutes. This duration is shorter than the minimum short operation duration constraint therefore only one short maintenance operation is created.

A search for a storage depot is performed. The depot nearest to the location of the maintenance operation is found at the beginning of Tr_135.

The optimal path is calculated from the initial location of the maintenance machine to the maintenance operation location and from the maintenance operation location to the storage depot.

A small extract of the tool text output describing the path is presented below:

```
Path(1) Name: Path starting at 0@Tr_628, doing work between 5193@Tr_134 and 15250@Tr_134,
ending at 0@Tr_135.
 -> 0 meters from track Tr_628
 -> 5570 meters from track Tr_628
 -> 6300 meters from track Tr_632
 -> 6100 meters from track Tr_634
 -> 9900 meters from track Tr_636
 ...
 -> 2296 meters from track Tr_3
 -> 0 meters from track Tr_30
 -> 5761 meters from track Tr_31
 -> 5193 meters from track Tr_134
 -> 15250 meters from track Tr_134
```

-> 0 meters from track Tr_135

A time-window shorter than 24 hours is generated, limiting the scheduling of the short maintenance operation to:

- Minimum Start Time: 10/01/2014, at 0:00 hours
- Maximum End Time: 10/01/2014, at 23:59 hours

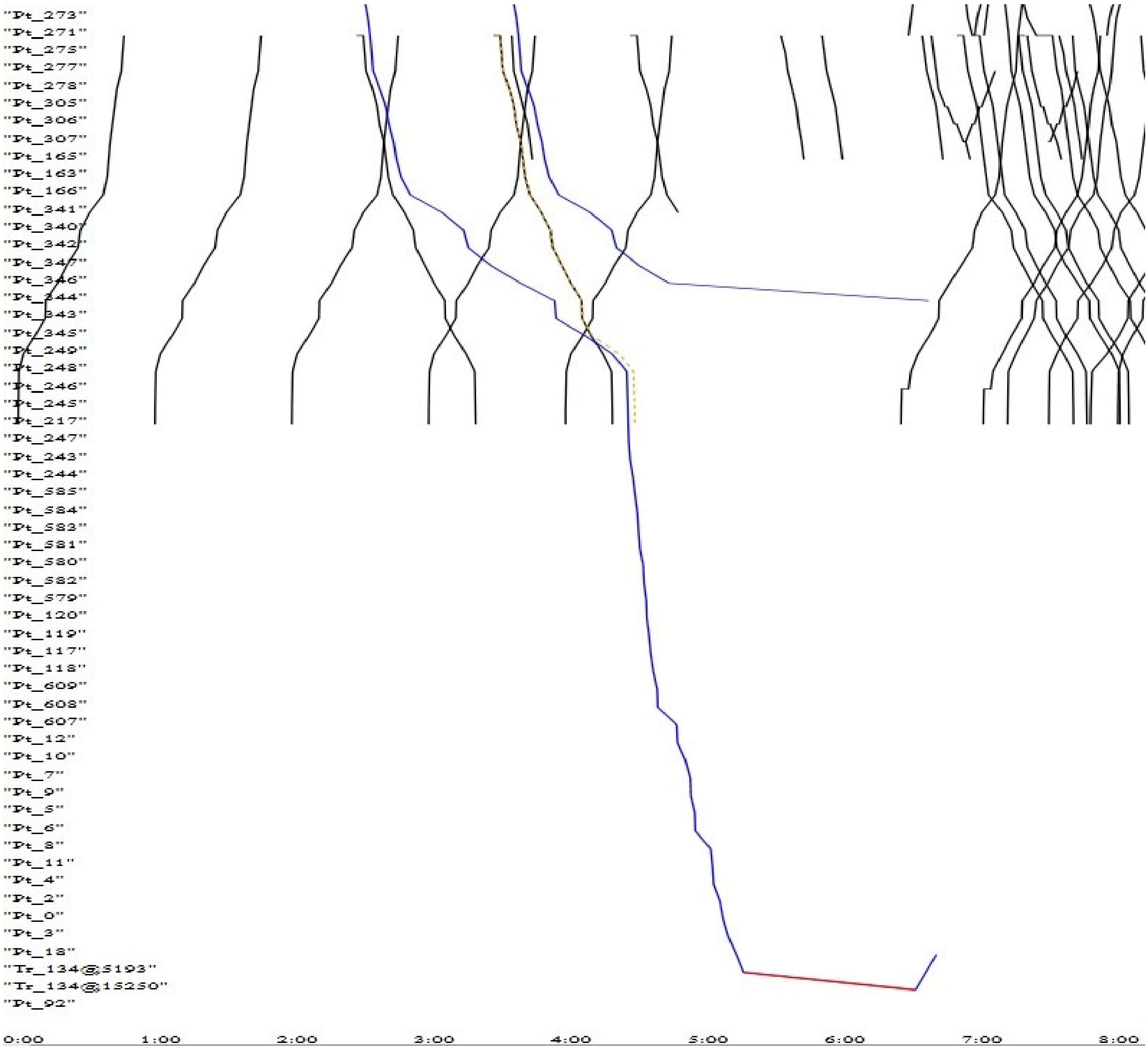
Finally, the TWIP module is used to fine-schedule the short maintenance operation.

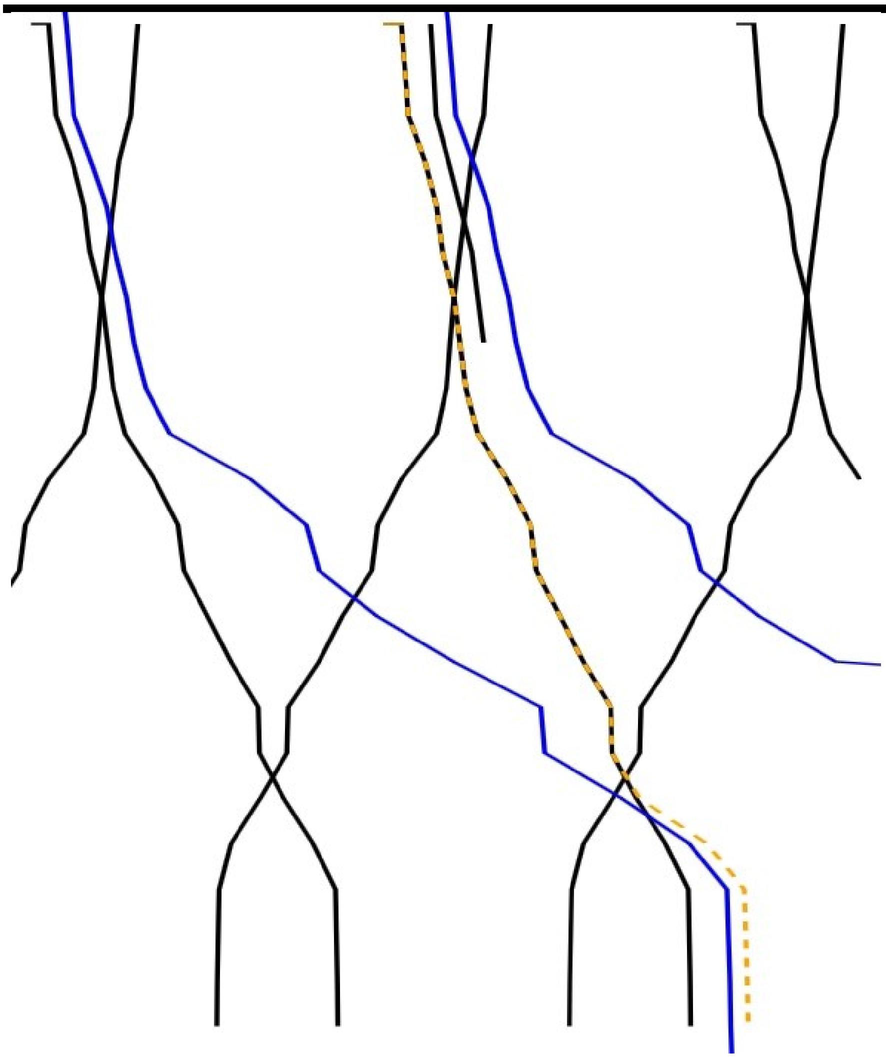
TWIP Input Data

The maintenance short operation is “Measurement_before_TampingOp_59”. It consists of a Measurement to do between locations 5193 and 15250 on track “Tr_134”. TWIP has to insert this operation on January 10th, 2014.

The maintenance train associated with this operation is “MeasurementTrain_4” (Travel Speed: 100 km/h; Work Speed: 8 km/h)

TWIP Graphical Output





TWIP Numerical results

- 0 train is cancelled
- 1 train is delayed (Tp_269_1)
- The maintenance train starts at 01:16:11 and ends at 06:49:09 (makespan : 05:32:58)
- The maintenance operation starts at 05:24:35 and ends at 06:40:00
- The solution's cost equals 40978

TWIP Results Analysis

This example concerns the same day as example 1.
The space-time diagram shows that the maintenance of example 1 has been approved by LTP. The associated maintenance train path is taken into account in the current optimisation.

In this example, TWIP cannot avoid delaying one commercial train. The yellow dotted curve corresponds to the delayed train.

For this example the TWIP module reported that the short maintenance operation scheduling was feasible and that no commercial train was cancelled.

In this case the validation process finalizes after fine-scheduling this short maintenance operation.

Composition of the TWG module output

The TWG module composes its output based on splitting of the maintenance operation, the calculated maintenance machine path and the scheduling results from the TWIP module.

For this example the output of the TWG module is composed of:

- A list of cancelled trains: No cancelled trains
- A list of delayed trains: One delayed train
- One short maintenance operation with the following schedule:
    - Operation Start: 10/01/2014, at 05:24:35
    - Operation End: 10/01/2014, at 06:40:00
- The full maintenance train path with a defined schedule.
- Objective value: 40978

This output is the value returned to the LTP module, when the TWG module is called by the LTP module.

Consideration of a parallel implementation of a least cost scheduling strategy

The current strategy used by the TWG module finds the earliest feasible solution in the granted time-window without cancelling trains. This strategy minimizes the usage of the TWIP module in order to achieve a faster execution time, but its results can be suboptimal in regards of impact on commercial train schedules.

A least cost strategy can be applied to find the solution with the least impact on commercial train schedules. In order to find a cost-optimal solution this strategy needs to call the TWIP module several times to find independent solutions along the whole time-window.

The least cost strategy would result in slow execution time if implemented as a serial computation, but the independent nature of the calculated solutions makes it possible to implement it as a parallel computation in an MPI grid.

The parallel least cost strategy is not implemented in the current version of TWG, and it is proposed as further work.

A test was performed for the present example, simulating a least cost strategy. The following cost optimal result was found:

- A list of cancelled trains: No cancelled trains
- A list of delayed trains: No delayed trains
- The solution is composed by one short maintenance operation:
  - Operation Start: 11/01/2014, at 04:14:45
  - Operation End: 11/01/2014, at 05:30:11
- The full maintenance train path with a defined schedule.
- Total cost: 40813

As seen in this result a solution with no delayed trains was found by the least cost strategy.

e) Validation of an operation – example 3

This test case corresponds to the tenth validation call performed by the LTP module.
The maintenance operation provided by the LTP module is fine-scheduled and validated in a joint effort of the TWG and TWIP modules.

The tool is launched with the following command line:

```
./automain LTP data/demo.xml data/refdata.xsd 5
```

In this case three commercial trains were delayed in order to schedule the maintenance operation. The example presents a situation with medium traffic load near to the maintenance operation location.

The validation process starts with a validation call from the LTP module to the TWG module.

The inputs received by the TWG module from the LTP module are:

- Reference data
- Assigned maintenance machine data, including:
  - Travel speed: 100 km/h
  - Work speed: 1.3 km/h
  - Initial location: at the beginning of Tr_628
- Maintenance operation data, including:
  - Start location of the operation: at 6664m from the beginning of Tr_612
  - End location of the operation: at 12469m from the beginning of Tr_612
  - Length of the operation: 5805m
- Scheduling time-window (medium-term):
  - Machine available date: 01/02/2014, at 0:00 hours
  - Target end date: 06/02/2014, at 0:00 hours
  - Latest end date: 08/02/2014, at 0:00 hours

The estimated duration of the maintenance operation is 4 hours 28 minutes. This duration leads to a balanced split of the maintenance operation into two short maintenance operations of 2 hours 14 minutes each.

The maintenance operation is split in only two short maintenance operations in order to comply with the minimum short maintenance operation duration constraint.

A search for a storage depot is performed. The depot nearest to the location of the maintenance operation is found at the beginning of Tr_611.

For the first short maintenance operation the optimal path is calculated from the initial location of the maintenance machine to the maintenance operation location and from the maintenance operation location to the storage depot.

A small extract of the tool text output describing the path is presented below:

```
Path(1) Name: Path starting at 0@Tr_628, doing work between 6664@Tr_612 and 9566@Tr_612,
ending at 0@Tr_611.
 -> 0 meters from track Tr_628
 -> 5570 meters from track Tr_628
 -> 6300 meters from track Tr_632
 -> 6100 meters from track Tr_634
 -> 9900 meters from track Tr_636
 ...
 -> 2105 meters from track Tr_536
 -> 3332 meters from track Tr_540
 -> 6800 meters from track Tr_271
 -> 6664 meters from track Tr_612
 -> 9566 meters from track Tr_612
 -> 0 meters from track Tr_611
```

A time-window shorter than 24 hours is generated, limiting the scheduling of the short maintenance operation to:

- Minimum Start Time: 01/02/2014, at 0:00 hours
- Maximum End Time: 01/02/2014, at 23:59 hours

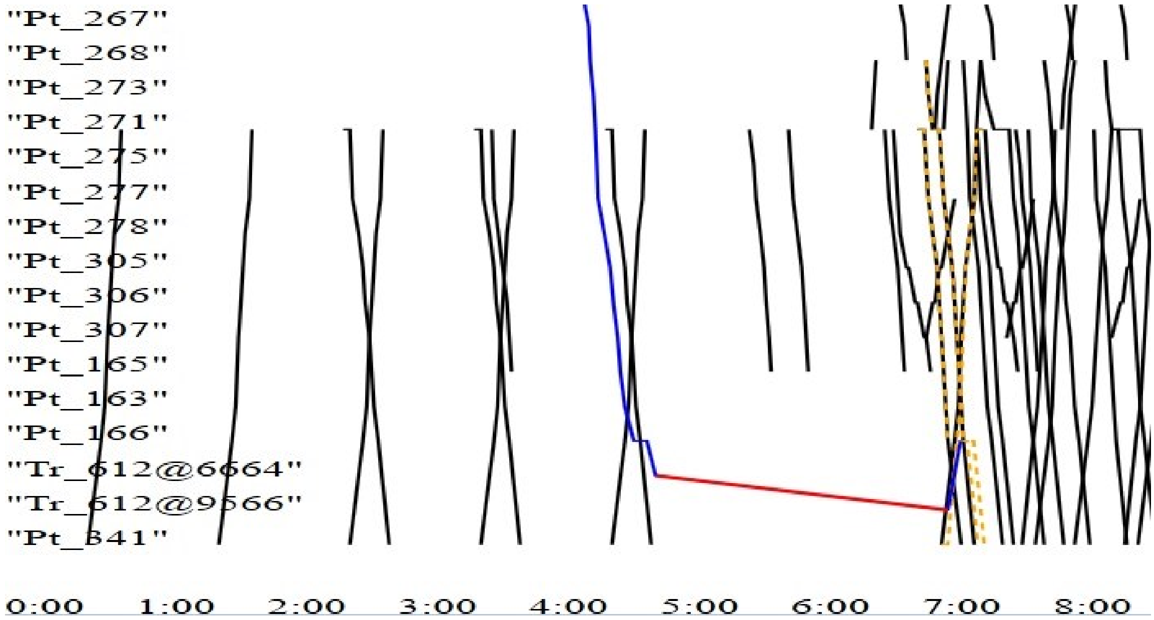
Finally, the TWIP module is used to fine-schedule the short maintenance operation.

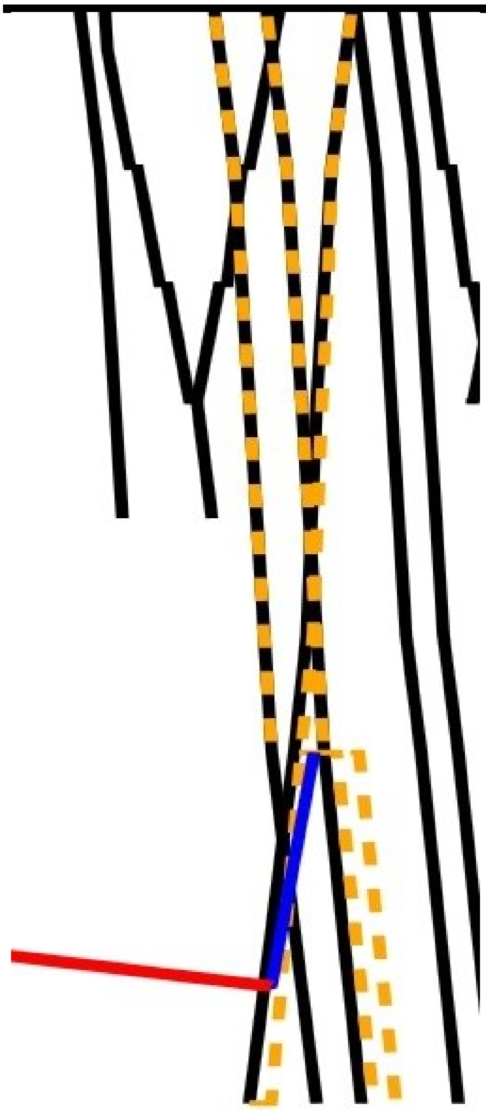
TWIP Input Data

The maintenance short operation is “GrindingOp_71”. It consists of a Grinding to do between locations 6664 and 9566 on track “Tr_612”. TWIP has to insert this operation on February 1st, 2014.

The maintenance train associated with this operation is “GrindingTrain_3” (Travel Speed: 100 km/h; Work Speed: 1.3 km/h)

TWIP Graphical Output





TWIP Numerical results

- 0 train is cancelled
- 3 trains are delayed (Tp_45, Tp_99, Tp_140)
- The maintenance train starts at 03:05:35 and ends at 07:17:41 (makespan 04:12:06)
- The maintenance operation starts at 04:58:00 and ends at 07:11:56 (duration 02:13:56)
- The solution's cost equals 31685

TWIP Results Analysis

In the first graphical output (general view), we can see that TWIP decides to use the siding of the maintenance train in order to allow the overtaking of the commercial train.

In the 2nd graphical output (zoomed view), the commercial trains stop in order to allow the maintenance train to continue on its way to the depot.

For this short maintenance operation the TWIP module reported that the scheduling was feasible and that no commercial train was cancelled.

After the first short maintenance operation is scheduled, the second short maintenance operation is processed by the TWG module.

First the optimal path is calculated from the storage depot to the maintenance operation location and from the maintenance operation location back to the storage depot.

The tool text output describing the path is presented below:

```
Path(1) Name: Path starting at 0@Tr_611, doing work between 9566@Tr_612 and 12469@Tr_612,
ending at 0@Tr_611.
 -> 0 meters from track Tr_611
 -> 9566 meters from track Tr_612
 -> 12469 meters from track Tr_612
 -> 0 meters from track Tr_611
```

A time-window shorter than 24 hours is generated, limiting the scheduling of the short maintenance operation to:

- Minimum Start Time: 02/02/2014, at 0:00 hours
- Maximum End Time: 02/02/2014, at 23:59 hours

Finally, the TWIP module is used to fine-schedule the short maintenance operation.

For this short maintenance operation the TWIP module reported that the scheduling was feasible and that no commercial train was cancelled.

In this case the validation process finalizes after fine-scheduling this short maintenance operation.

Composition of the TWG module output

The TWG module composes its output based on splitting of the maintenance operation, the calculated maintenance machine path and the scheduling results from the TWIP module.

For this example the output of the TWG module is composed of:

- A list of cancelled trains: No cancelled trains
- A list of delayed trains: Three delayed trains
- Two short maintenance operation with the following schedules:
    - First short maintenance operation:
        - Operation Start: 01/02/2014, at 04:58:00
        - Operation End: 01/02/2014, at 07:11:56
    - Second short maintenance operation:
        - Operation Start: 02/02/2014, at 00:05:44

    - Operation End: 02/02/2014, at 02:19:43
- The full maintenance train path with a defined schedule.
- Total cost: 49349

This output is the value returned to the LTP module, when the TWG module is called by the LTP module.

Comments on the second solution provided by TWG

In this example the *Latest end date* and the *Target end date* are not equal; therefore there are two different input time-windows and two solutions are calculated by the TWG module.

The two input time-windows have the same start date, but different end dates. The first time-window is longer than the second time-window and the additional days could be used to obtain a solution with less impact on the commercial train schedule.

Nevertheless, as explained during section III.d, the current strategy used by TWG finds the earliest feasible solution inside the granted time-window without cancelling trains and does not check for additional solutions.

As the two input time-windows have the same start date and are long enough to contain at least one feasible solution, this solution is the earliest feasible solution. The two solutions calculated by the TWG module are identical in this case.

Results of the simulation of a least cost scheduling strategy

A test was performed for the present example, simulating a least cost strategy. The following cost optimal result was found:

- A list of cancelled trains: No cancelled trains
- A list of delayed trains: No delayed trains
- The solution is composed by one short maintenance operation:
    - Operation Start: 02/02/2014, at 01:46:25
    - Operation End: 02/02/2014, at 06:14:21
- The full maintenance train path with a defined schedule.
- Total cost: 45819

As seen in this result a solution with no delayed trains was found by the least cost strategy, performing the whole maintenance operation in one day.

f) Insertion of an operation into an existing planning

In this section, we show how the algorithms embedded in the tool enable a maintenance operation to be inserted into an existing planning. A typical application of this is the situation where a measurement train detects a defect to be corrected while a maintenance planning is already in being executed.

The command line used in this case is the following:

```
$ ./automain LTP data/demo.xml data/refdataxml.xsd 0 noValidation insertionMode
```

Where

- ./automain is the name of the tool (executable file),
- LTP the module called,
- data/demo.xml the input data file, containing both the operations of the existing planning, and the additional operation to be inserted,
- data/refdataxml.xsd the xml schema document file,
- 0 the time, in minutes, allowed to LTP to build the existing planning,
- noValidation is a flag used to skip validation of operations by TWG and TWIP,
- insertionMode is a flag used to trigger the insertion mode of the tool.

Once launched, the tool starts processing the data and running the algorithms:

```
19/12/13 23:05:23 INFO AUTOMAIN CALLED WITH THE FOLLOWING COMMAND LINE:
19/12/13 23:05:23 INFO ./automain LTP data/demo.xml data/refdataxml.xsd 0 noValidation insertionMode
19/12/13 23:05:23 INFO ************************************
19/12/13 23:05:23 INFO data/demo.xml validates data/refdataxml.xsd
19/12/13 23:05:23 INFO Reading data/demo.xml...
19/12/13 23:05:36 INFO Nb depot = 721
19/12/13 23:05:36 INFO Nb location = 8400
19/12/13 23:05:36 INFO Nb point = 704
19/12/13 23:05:36 INFO Nb track = 1274
19/12/13 23:05:36 INFO Nb trackGroup = 788
19/12/13 23:05:36 INFO Nb train = 277
19/12/13 23:05:36 INFO Mean nb etp per train = 53.0505
19/12/13 23:05:36 INFO Nb maintenance op = 200
19/12/13 23:05:36 INFO Total length of tracks (m) = 6693878
19/12/13 23:05:36 INFO File data/demo.xml was read.
19/12/13 23:05:36 INFO ************************************
19/12/13 23:05:36 INFO Graph is connected: explored 704 points.
19/12/13 23:05:36 INFO Calling LTP...
19/12/13 23:05:36    DEBUG ###### Predecessors ######
19/12/13 23:05:36    DEBUG Ope Measurement_before_TampingOp_0 predecessor of Ope TampingOp_0
19/12/13 23:05:36    DEBUG Ope Measurement_before_TampingOp_1 predecessor of Ope TampingOp_1
19/12/13 23:05:36    DEBUG Ope Measurement_before_TampingOp_10 predecessor of Ope TampingOp_10
19/12/13 23:05:36    DEBUG Ope Measurement_before_TampingOp_100 predecessor of Ope TampingOp_100
19/12/13 23:05:36    DEBUG Ope Measurement_before_TampingOp_101 predecessor of Ope TampingOp_101
19/12/13 23:05:36    DEBUG Ope Measurement_before_TampingOp_102 predecessor of Ope TampingOp_102
[...]
19/12/13 23:05:36    DEBUG Ope TampingOp_81 and Ope TampingOp_86 aren't compatible.
19/12/13 23:05:36    DEBUG Ope TampingOp_83 and Ope TampingOp_93 aren't compatible.
19/12/13 23:05:36    DEBUG ####################################
19/12/13 23:05:36   ALGO Starting planning build...
19/12/13 23:05:37   ALGO Planning built ok.
19/12/13 23:05:37   ALGO Planning constraints OK.
19/12/13 23:05:37   ALGO Planning cost : 499648
19/12/13 23:05:37   ALGO Possession cost : 447000
19/12/13 23:05:37   ALGO Lower bound for possession cost : 217000
19/12/13 23:05:37   ALGO Starting simulated annealing...
19/12/13 23:05:38   ALGO Simulated annealing ok.
19/12/13 23:05:38   ALGO Planning constraints OK.
19/12/13 23:05:38   ALGO Planning cost : 499648
19/12/13 23:05:38   ALGO Possession cost : 447000
19/12/13 23:05:38 INFO Planning exported to out/ltp.csv
Press ENTER to insert operation TampingOp_98 into the existing planning.
```

Once the existing planning is built, the tool asks to press ENTER to trigger the insertion of a tamping operation ("TampingOp_98", a tamping operation of 1782 meters, whose target date is Nov 2nd, and whose deadline is Nov 30th) into the existing planning.

When the order is given by the user, the tool has to find the best suitable machine and the best suitable dates to perform the insertion of this operation. This is done very quickly (less than a second), without changing the already existing planning. The tool proposes the insertion of "TampingOp_98" on Tamping train n°1, and identifies Nov 30th as an appropriate date to start the operation. The Gantt charts associated with the existing planning and with the insertion are shown in Figure 5 and Figure 6 respectively.

insertOperation : inserted TampingOp_98 on TampingTrain_1 on [2014-Nov-28, 2014-Nov-30, 2014-Dec-01]
19/12/13 23:07:33 INFO Planning exported to out/afterInsertion.csv
19/12/13 23:07:33 INFO LTP was called.

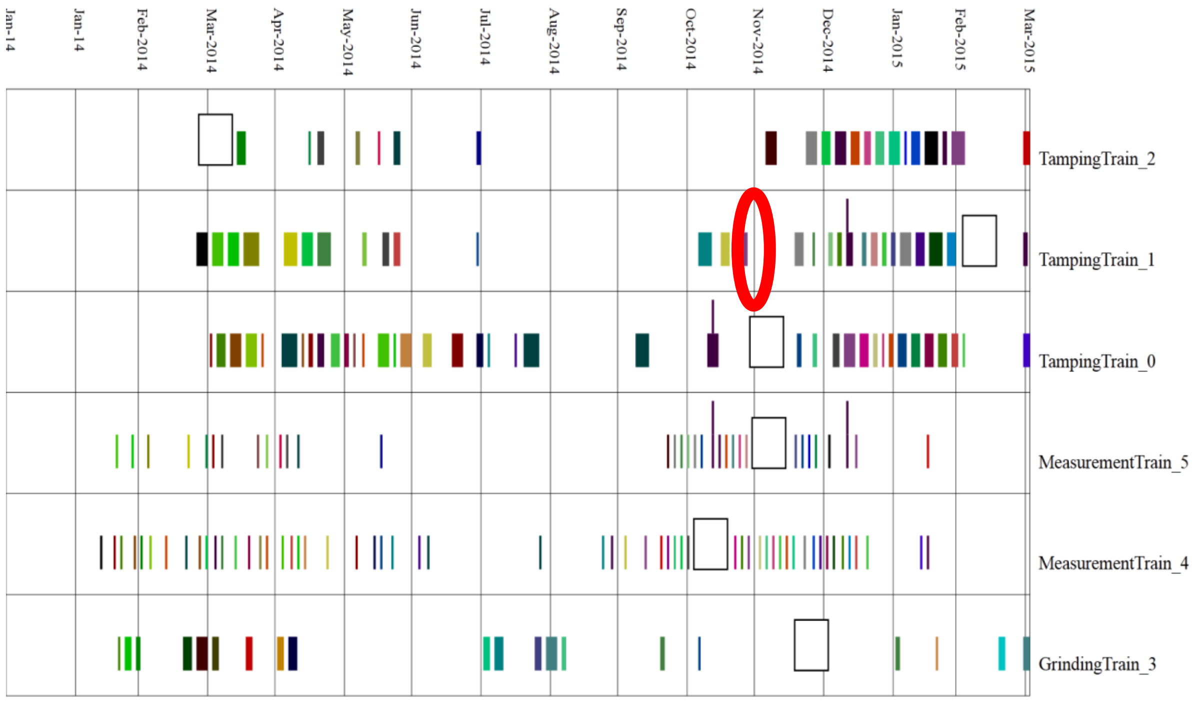


Figure 5 - Existing planning

This assignment to tamping train n°1 is feasible (the type of machine matches the type required for the operation), and this resource is available to travel from its previous location on Nov 28th, do the maintenance work on Nov 30th and Dec 1st, and has enough time to reach its next operation.

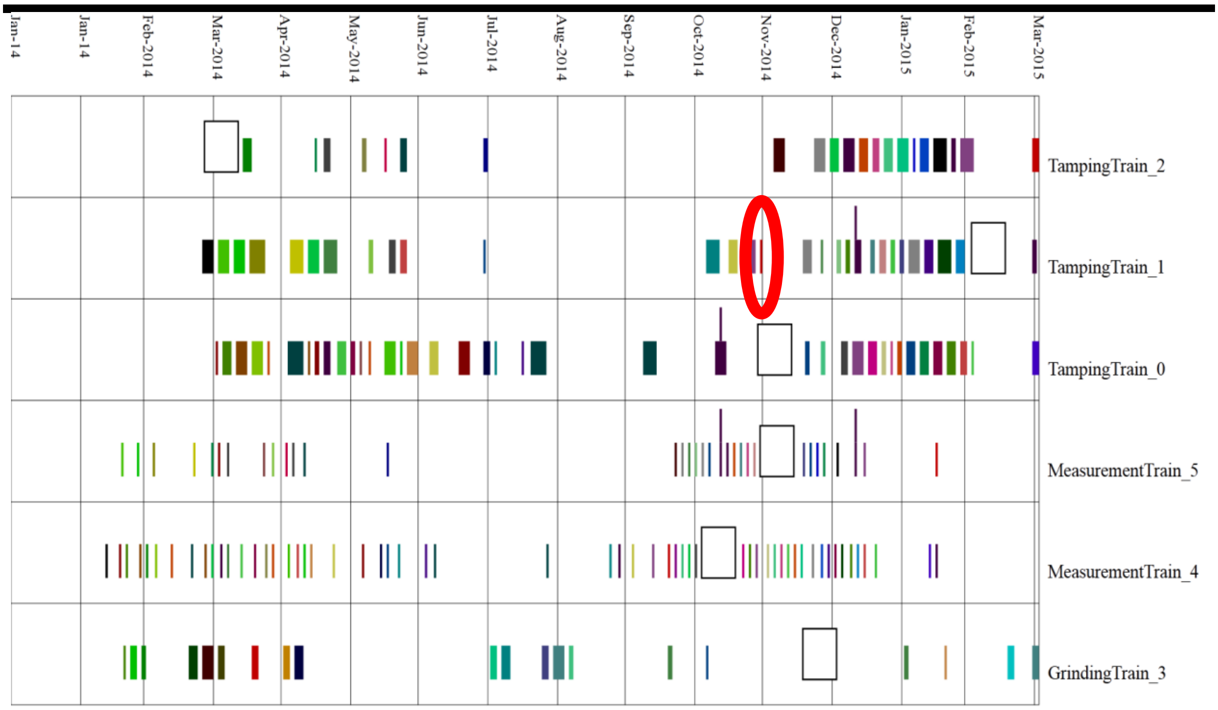


Figure 6 - Insertion of a tamping operation into an existing planning

IV. Evaluation of WP3 and WP4 results

a) *High-speed maintenance evaluation*

In this section, we try to evaluate the improvements brought by WP4 on various parameters influencing the possession time required to perform the maintenance on tracks. The variety of these improvements being large, we chose to focus on a few examples which are considered most significant. In this case, we selected the following improvements:

- WP4 – improvement 1 (instance denoted WP4.1): Increase of tamping speed by 29%, corresponding to the potential for improvement on “Twice tamping, 2 sleeper movement” in its pessimistic version).
- WP4 – improvement 2 (instance denoted WP4.2): Increase of tamping speed by 52%, corresponding to the potential for improvement on “Standard method” in its pessimistic version).
- WP4 – improvement 3 (instance denoted WP4.3): Increase of measurement speed by 40%, corresponding to the potential for improvement on “track recording car”).

Instance	Improvement	Possession time without optimization (days)	Gain without optimization	Possession time after optimization (days)	Gain after 20 min optimization by LTP
Base instance	None	442	0%	405	8%

WP4.1	Tamping speed improved by 29%	389	12%	352	20%
WP4.2	Tamping speed improved by 52%	366	17%	332	25%
WP4.3	Measurement speed improved by 40%	442	0%	405	8%

Table 1 - Gains of WP3 and WP4

Table 1 presents the possession time measured by the tool using the same dataset as a base ("base instance") and modifying the characteristics of maintenance machines according to the improvement to be tested. This is done for each improvement, evaluating successively *without* optimization (i.e. applying identical basic planning rules) and *after 20 min optimization* by LTP algorithms. So, for each improvement, the gain in possession time is evaluated for the improvement itself, and for the improvement combined with optimized planning algorithms, compared with the reference value obtained without optimization on the base instance.

The results measured show that increasing tamping speed has a significant impact on possession time (12% and 17% for WP4.1 and WP4.2 respectively). This is quite logical as tamping operations represent a high proportion of the total number of maintenance operations. However, increasing measurement speed does not lead to a gain in possession time measured by the tool. This can be explained by the unit used to measure possession time which is the number of days where possessions occur, and which is too imprecise to capture reductions of less than a day. Indeed, as the speed of measurement is much higher than the speed of other operations, measurement operations are usually completed within less than a day, even without improvement. So, this result does not mean that improving measurement speed has no impact on possession time; it only means that the long-term planning of the tool is not designed to represent such small variations in possession time within given days. A model with a greater precision level is therefore necessary to highlight these types of gains. Nevertheless, the associated optimization problem would require greater computation times as well, unless algorithms are improved.

When these improvements are combined with optimization algorithms, the gains are not surprisingly larger. They rise to 20%, 25%, and 8% (for WP4.1, WP4.2, and WP4.3, respectively). An interesting aspect to point out is the stability of the gain brought by optimization: over all instances, the additional gain is evaluated around 8% (12→20% for WP4.1, 17→25% for WP4.2, 0→8% for WP4.3). The number of instances is too low to formally conclude but it seems to indicate that the algorithms are quite robust with respect to the input data on machine speeds.

b) High-speed inspection evaluation

As discussed in the previous section, due the design of the tool and the levels of details used in the different modules, the current version of the tool does not enable to precisely measure gains in possession time associated with improvements in track measurement.

We consider that these gains associated with the duration of individual possessions could be more precisely assessed by modifying some functionalities of the tool, as part of post-project improvements. This is discussed in Section V-b) “Identified potential improvements”.

V. Conclusion

a) Main outcomes of WP5

In this section, we focus on two features of the tool developed which we consider as the most representative: the integration of global and local considerations, and the capability to optimize different criteria.

Integrate both global planning and local scheduling of maintenance operations

The algorithms embedded in our tool enable different levels of planning and scheduling to be synchronised. Indeed, both macrosopic and microscopic aspects are integrated in the same approach. To the best of our knowledge, this integration is something new, developed within WP5 of the AUTOMAIN project.

The need to handle different levels is quite clear. Indeed, scheduling individually each maintenance operation without a global view might lead to machine unavailability or incompatible operations being planned simultaneously. However, not taking into account particularities of operations at a more detailed level may also result in conflicts with other trains or unfeasible task times.

The macroscopic vision consists in having a global optimization of a fleet of maintenance machines, such as measurement, grinding and tamping machines, and defining for each of them the planned activity over a time horizon of several months or years. Obviously, to be feasible in practice, sufficient travel time should be allowed between successive operations of a single machine. It is also necessary to make sure that incompatible operations are not performed at the same time, and that no operations are planned during the machine's own maintenance. Another aspect handled at this level is the need to perform operations in a certain order. For instance, a measurement operation might be required at least a few weeks prior to a tamping operation. This kind of inter-operation constraint must be integrated to make sure that the resulting optimized planning will be feasible. Finally, the optimization of possession time requires having a global vision on all operations, in order to identify those which can be combined during the same track possessions.

On the other hand, microscopic constraints must be considered for each maintenance operation. The most significant of these considerations is, naturally, to integrate maintenance within the commercial traffic. In this regard, different strategies can be explored by the tool. It can either give priority to traffic, and find the best way to schedule maintenance during the less busy periods, or give priority to maintenance and find the best way to minimize impact on commercial traffic. This is done taking into account the expected commercial timetable and the local network topology.

The link between global and local is obtained by specific algorithms which are able to split large operations over several possessions, and which find the optimal routes for machines over the network between operations. These algorithms have to deal both with the global planning aspects and with local considerations. Consequently, our tool is composed of several modules dedicated to these specific problems, communicating their results to each other to ultimately provide a global planning defining when and how each maintenance operation is performed.

Capability to optimize different criteria

The tool developed within WP5 of AUTOMAIN is able to deal with objectives of different natures. For instance, it can optimize either the capacity of the network, by minimizing track possession times, or machine usage, by minimizing the number of km travelled between operations, or even find compromises between these criteria.

This feature is an originality of the tool, as most studies focus on optimizing network capacity only. This was achieved by a careful analysis and tuning of the algorithms, in order to find suitable parameters influencing the search procedure within the solution space.

Optimizing network capacity is made possible by combining several operations into a single possession. Obviously, these operations must be compatible with each other when they are combined. The goal here is to identify such compatible operations, and find common time periods where they could be performed simultaneously. If tests are performed on a broader range of situations, early experiments showed that the reduction in possession time obtained by the optimization algorithms developed could reach up to 14% compared with basic planning rules.

A maintenance planning of machines over a time horizon of a few months generates a lot of travel from the end of an operation towards the beginning of the next operation planned on the same machine. Depending on the sequence of operations for each machine, the total distance travelled by machines can vary greatly. Intuitively, the ideal situation would be to perform operations geographically close to each other during the same time period. Given the deadlines potentially imposed on operations, this is not always possible. Moreover, the computational complexity of this type of problem is known to be very high. Even with only one machine, the well-known Travelling Salesman Problem (TSP) has a high theoretical complexity. For these reasons, dedicated algorithms were developed and applied successfully. The first results show that using these types of algorithms to minimize distance travelled can save up to 69% if full priority is given to this criterion, or 42% if possession time is also optimized.

Hence, depending on the needs and on the user, the tool developed within WP5 of AUTOMAIN is able to optimize criteria of different natures. One major advantage over other tools is that it can handle them simultaneously and can thus bring together the different approaches of the different stakeholders of the maintenance business, in particular infrastructure managers and machine owners. Indeed, the objectives of these stakeholders are often different, resulting in negotiations and iterations to build together acceptable planning solutions. The cooperation between stakeholders could be improved by using these types of algorithms because they can find good compromises, optimizing their respective performance indicators. They can also help to estimate the best possible solutions for each and make everyone aware of the global gains of finding good compromises. This tool provides a holistic view of all aspects involved the tension between maintenance and traffic.

b) Achievement of initial objectives

The initial objectives of WP5 were described in the DoW (Annex I – "Description of Work" of the grant agreement) of the project in the following terms :

> *WP5 will work to develop operations research methods and tools for autonomous maintenance planning and scheduling. The planning problem identifies the time periods when a track segment should be closed for maintenance and the maintenance tasks that are to be undertaken. The scheduling (or timetabling) problem considers finding the start and completion times of each maintenance operation.*
>
> *Using the requirements developed in WP and WP2, a tool will be developed that takes inputs from a number of parameters and constraints (component state, traffic, required operations, maintenance time windows, resource availability…). This tool will be used to evaluate scenarios defined by Task 2.2 in order to evaluate the capacity and cost savings that could be attained by the innovations proposed in WP3 and 4.*

Clearly, the objectives of WP5 have been achieved. Indeed, optimization methods based on operations research techniques have been developed to tackle both track maintenance planning and scheduling problems. Moreover, a software prototype has been produced based on these methods, taking as input a number of information regarding network, traffic, maintenance, and automatically generating optimized planning and scheduling solutions. It was used to evaluate innovations proposed in other work packages.

c) Identified potential improvements

Even if the actual version is functional, there are several ways of making improvements.

Functional improvements

The functional improvements allow to access to new functionalities, or access to already existing functionalities in a different manner.

Taking into account already existing features

TWIP already takes into account some unused elements of Reference Data. If these elements are used in the future, it will require only a small amount of work to improve the results. The elements are the following:

- Several Maintenance train paths can be sent to TWIP instead of only one. During the optimisation TWIP would select the optimal one.
- For a time window, TWIP is able to deal with minimum start time, maximum start time, minimum end, time, maximum end time, instead of just start time and end time. This allows time intervals to be managed instead of fixed times.
- If there are several ways to perform a maintenance operation (for example, perform the whole operation, or split it into sub operations), called "modes", TWIP is able to take into account all of these modes and to find the best one (it is also possible to indicate a preference using a "mode cost" integrated to the objective function).
- It is possible to take into account maintenance tasks that begin on one track and finish on another track.

Some information existing in Reference Data is not used for the moment but can be easily integrated:

- It is possible to use two different speeds for the maintenance train according to its travelling direction.

Insert a maintenance task in an already optimized planning

When a planning is optimized by Automain, the results are exported in a XML document (out.xml) having the same format as the input data file. If one needs to insert a new maintenance task into the already existing planning, it is possible to initialise Automain using out.xml as an input data.

Take into account the security constraint relative to already fixed maintenance

When optimizing a maintenance task, some security constraints are taken into account in order to make the commercial trains travel more slowly when they are close to a maintenance task. However, in the case of an already existing maintenance train, this security constraint is not usable because only the maintenance train path is taken into account, and not the associated maintenance task.

Take into account punctual maintenance

The maintenance tasks considered in Automain take place between two locations separated by a given distance. The maintenance duration is computed using this distance and the maintenance speed. A punctual maintenance (i.e. same beginning and end locations) cannot be handled by the current version because the distance is nil. We can remedy this by considering that the maintenance is not punctual but that it is between very close locations.

Tuning on « no impact on commercial traffic » mode:
The “no impact on commercial traffic” mode was introduced very late in the project, and even if it already works, some fine tuning will be required to make it perfectly complementary to the other features.

Take into account regulatory and set-up times
In the current version of the prototype, only the core duration of maintenance activities are taken into account. However, depending on the type of work to perform, additional times are required before and after the actual maintenance works. These times are due to regulations and set-up / warm-up of the machines. Taking these times into account would not require heavy modification of the tool as the concept of inserting time windows into an existing grid remains unchanged.

Refine the possession time evaluation
As discussed in Section IV-b), the current version of the tool does not enable to precisely measure reductions of possession times associated with individual possessions. This is due to different units being used by the different modules of the tool. This refinement could be achieved by homogenizing the way possession time is modeled in the different modules constituting the tool.

Performance improvements

Sometimes a stakeholder may prefer to have a good solution quickly, rather than the best possible solution. There are several improvements which are possible:

Stop computing before the end

Stop computing when the solution is good enough
As the process of computing the best solution is quite long, it is possible to decide to stop the computing before the end, using criteria called “gap”, representing the difference between the current best solutions and the current best bound. Sometimes it is not necessary to have the best possible solution, especially if this solution is very long to compute. A solution having a small gap can also be a very good solution.

Stop computing after a given time
In the same way, it is possible to configure the solver to stop after a given time if a solution which is at least feasible is found.

Parallelize the optimization
TWG launches TWIP several times with different inputs. It is possible to parallelize Automain in order to have several TWIP instances running in parallel (with MPI for example).

d) Future perspectives

Under the Horizon 2020 Framework, the Institute for Quality, Safety and Transportation (iQST) is interested in the development of an industrial tool for planning and scheduling of railroad infrastructure maintenance and replacement strategies for old infrastructure. The Institut für Verkehrssicherheit und Automatisierungstechnik (iVA) would be an active partner in this development. The possibility of using the planning tool prototype developed within WP5 of AUTOMAIN as a base for further development of an industrial tool would be the responsibility of Dr. Quiroga in iQST.