Rerouting algorithm outline

This note briefly describes the rerouting algorithm to be implemented in TMS in order to support a dynamic modification of train routing when it is beneficial for scheduling improvement

Definitions and aims

With the term rerouting we intend a modification of the composed route assigned to a train inside a PPLG by replacing it with a different partial route sequences chosen from a give set of alternatives, possibly with a priority order. The alternative routes may share the same end points of the original one, but this is not a mandatory constraint, that means a rerouting action in a PPLG could imply route modification outside the PPLG itself, if the new end points are different.

The final aim of a rerouting action is to cope with conflicts that could not be solved by acting on train ordering only, and so to improve traffic fluidity and to reduce delays.

The role of rerouting is dramatically evident in stations, where fixed platform occupation policy limits performances and causes delays when the platform assigned to a train is occupied by another one, but there are empty platforms in the same station.

Rerouting constraints

A generic rerouting algorithms must keep in account a set of constraint arising from both railway practice and computational issues. We summarize them in the following.

Rerouting can be applied only within a definite time window, because

- it is not possible to modify the train routes too close in time and space to the current train position (physical constraints on route setting, time needed to inform the public)
- it is not useful to modify the train route too far in time and space, because the solution based on long-time forecasting could not be optimal in a dynamically changing environment.

Rerouting can be performed keeping in account train relations, that means when a train is re-routed, the correlated train must be re-routed too in order to maintain the relation. This is valid for all kind of relations, including the hard rolling stock relations.

The same train cannot be rerouted many times, in order to avoid confusion. So when a rerouting action has been decided for a given train in a given PPLG, such decision must be considered as definitive.

The search for the best rerouting action is a complex optimization task with a huge search space to be explored, even with a small set of alternatives. Exhaustive strategies are obviously not feasible, and only sub-optimal algorithms must be considered.

The proposed rerouting approach

In order to include the rerouting capability in the TMS without affecting too much the computational performances of the system, we propose an approach in which, at every new plan generation, a few rerouting alternatives are explored, in order to look for a better plan even if it is not the best in general. We exploit the fact that new plans are generated quite often, due to mismatches between planned schedules and real time operation, so we can distribute the computational extra effort needed for the rerouting analysis in different activation of the scheduling module (CRS1).

The main idea, at each new plan generation, is to look for all typical delay patterns in which a train A induces a heavy delay over another train B, as represented by an alternative arc between A and B whose activation causes a large variation in passing times of B. When A and B routings are fixed, such a delay is unavoidable, but it is possible that the delay may be reduced if routing of B could be changed.

So we have to define:

1. which couple of train A-B present a high delay pattern
2. which alternative route for B can be chosen as a good candidate

Answering to such questions, we can build a new graph and to perform a new plan generation as usual, obtaining a “alternative plan”. Than we can compare the alternative plan with the original one, by a suitable cost function, and to select the best one. The cost function can be simply the total delay or a most complicated function like punctuality (e.g. the percentage of trains with delay lower than a given threshold).

In order to avoid wasting computational time, it is highly opportune that TMS keeps track of all explored alternatives, even if they are not beneficial, in order to focus its attention elsewhere in the successive searches. This can be done using a taboo list in which to store all unsuccessful tests, while the successful ones become part of the current schedule.

The implementation of the proposed rerouting approach relies on the definition of two new heuristics. The first one to select the candidate trains and the location (e.g. the block) for the intervention, the second one to select the alternative route to be tested. Such heuristics will be briefly outlined in the following.

Flow chart of the proposed approach

The main flow chart of the proposed approach is reported in Fig.1. Here we can note the process is composed by a cycle of graph building and optimization tasks, repeated every time with a different choice of alternative routes as produced by the rerouting check task. The cycle can be stopped depending on available computing resources, by fixing the maximum number of iterations. Each iteration produces a new alternative plan including a different re-routing action.

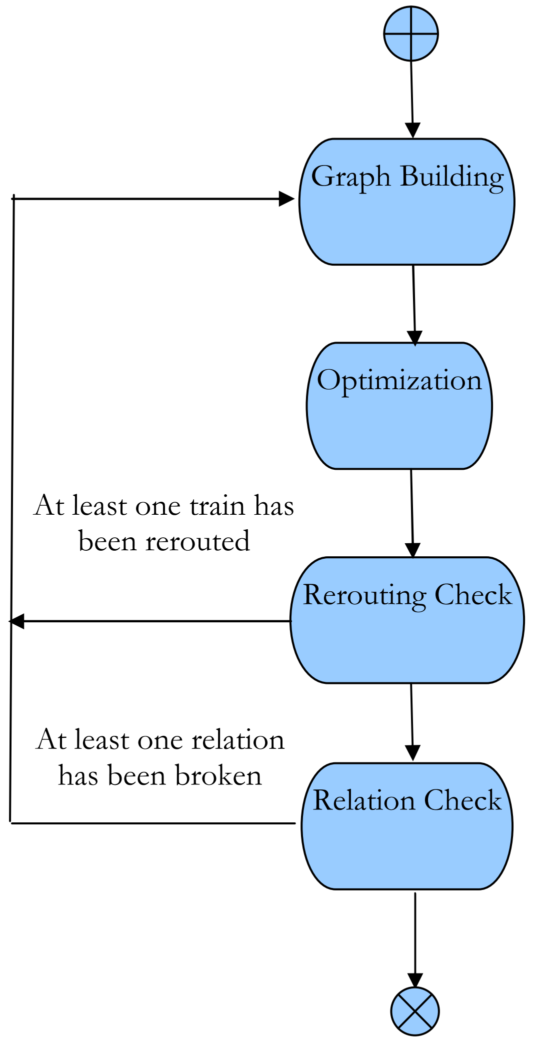


Fig.1 – High level sketch of the rerouting process

The rerouting check task is the key issue of the full process. It must select the train and the route to be tested, by implementing the two heuristics cited above. Its detailed flow chart is shown in Fig.2.

Looking at the figure, the main point to be emphasized are:

- the train rerouting check, aiming to select the delayed train which is most suitable to be re-routed;
- the selection of the best alternative route.

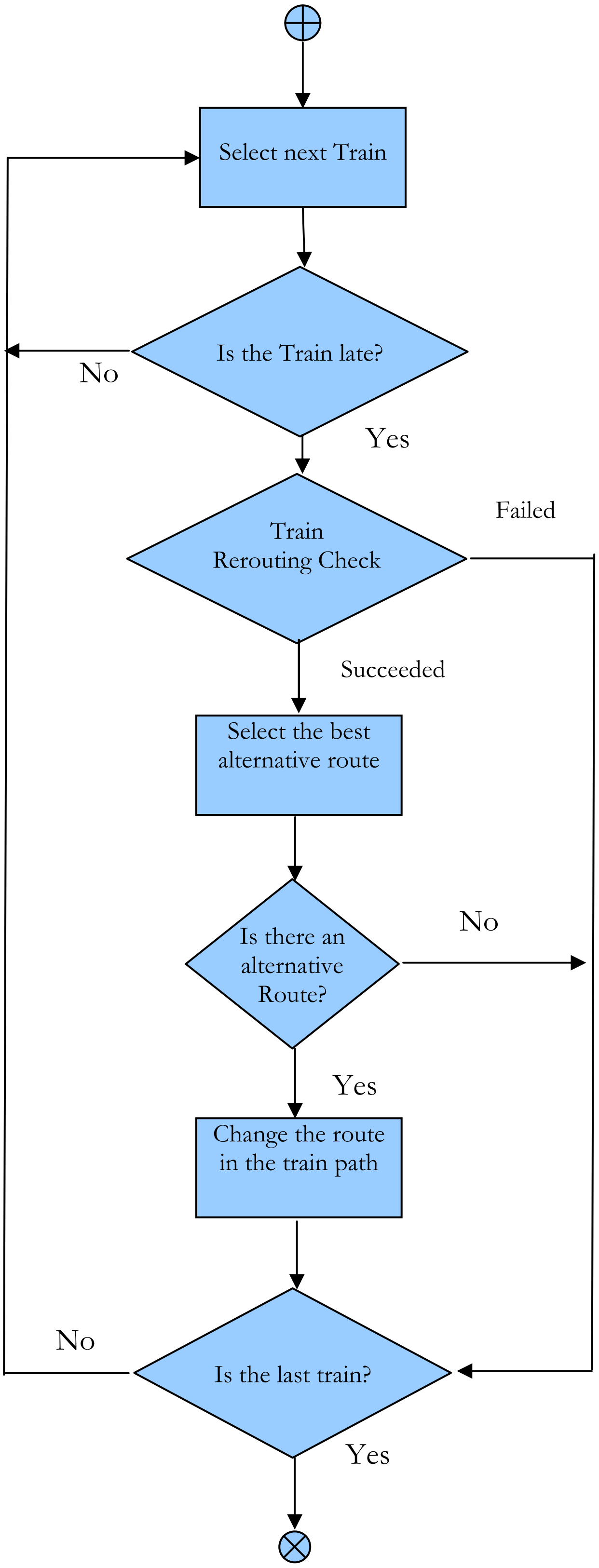
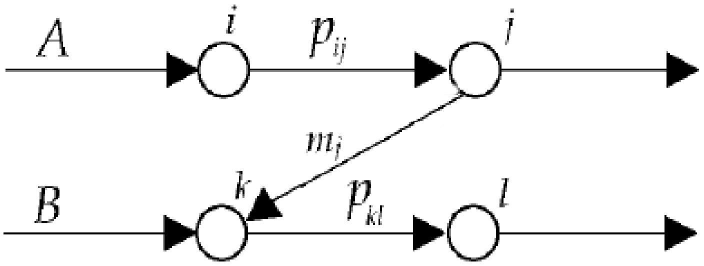


Fig.2 – Detailed sketch of the rerouting check

Train rerouting check heuristic

The train rerouting check is based on the evaluation of the added delay on a train graph node, that means the delay added on train B by a train A which shares a bock with B and is in advance. In normal flow, such added delay is zero, and it grows when hindering occurs. In some case, for instance at stops, the train B may suffer for high added delay caused by A, for instance when B has to wait for a platform currently occupied by A.



In the alternative graph formalism, the added delay imposed by the train B over the train A at node k, is written as:

$$D_k = max(0,T_j+m_j-T_k)$$

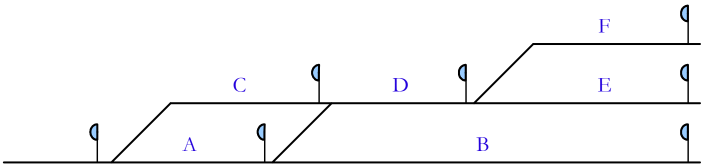
So, the proposed heuristic selects, scanning each delayed train, the arc producing the maximum added delay. This identify a conflict and the block section to be avoided (if possible) by a rerouting action.

Then, the selected arc for successive analysis will be the one having maximum added delay and pointing to a delayed and re-routable train. Let us remark that it is not the most critical arc, but the one that can be avoided by a rerouting action.

Alternative route selection heuristic

The selection of the best alternative route to be tested is performed by looking at the alternative route minimizing the occurrences of other trains in the same PPLG in a given time window (e.g. a few minutes) around the conflict time. So we look for all trains passing through the same PPLG of the conflict in this time window and evaluate a numeric score for each alternative route simply by counting how many switches are in common between this route and the routes already used by other trains (which are not rerouted). The aim is to choose the alternative route minimizing possible train interactions.

Let we clarify this procedure with an example, considering the entrance section of a PPLG as in Fig.3, and a given set of alternative routes for 3 trains



Train	Composed Route
1	A – B
2	A - B
3	A – D - E

Alt_Route_Set
A - B
A – D – E
C – D – E

Fig.3 – Entrance section of a PPLG with alternative routes

When train 1 is delayed, train 2 has to wait for the A-B route. Having a set of alternative routes for A-B, a rerouting of train 2 could be beneficial. The selected alternative will be the one minimizing interactions with other trains. There is another train passing through the PPLG in the same time window, using route A-D-E, so the best choice (avoiding A-B) will be C-D-F.

Conclusions

After such train and route selections, a new graph can be built and we can produce a new plan by the standard graph optimization in CRS1. It remains only to decide whether the new plan is better than the old one or not, using a cost function like the total delay. Let us note that, at this stage, it is possible to use all kind of function which are computable on the base of the time schedule (e.g. the punctuality).
This is not the same case of graph optimisation, in which the possible cost functions are constrained by the structure of the graph. Now we have only to compare two complete plans, and so we have no constraint about the kind of comparison to be done.

Finally, let us remind that, being completely based upon heuristics, it is clear that such rerouting procedures must be tuned over a set of test cased, in order to improve the selection of the most promising set of train and routes to be analyzed.
However, operating on a single rerouting test at each scheduling activation, we ensure to keep stable the routing of most trains, with small modifications widely distributed in time.