The Netherlands
Research School for
Transport, Infrastructure
and Logistics

TRAIL

11th TRAIL Congress
November 2010

# BI-OBJECTIVE CONFLICT DETECTION AND RESOLUTION

## Minimizing train delays and maximizing passenger connections

**Francesco Corman MSc[1], Dr. Andrea D'Ariano[2],**
**Prof. dr. Dario Pacciarelli[2], Dr. Marco Pranzo[3]**
[1]Faculty of Civil Engineering and Geosciences, Department of Transport and Planning,
Delft University of Technology, the Netherlands
[2]Dipartimento di Informatica e Automazione, Università degli Studi Roma Tre, Italy
[3]Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Siena, Italy

# ABSTRACT

Railway timetables define routes, orders and timings for all trains running in the network. Usually, timetables provide good connectivity between different train services for a number of origins and destinations. For each pair of connected train services, the waiting train is scheduled to depart sufficiently later with respect to its feeder train in order to allow the movement of passengers from one train to the other. During operations, train traffic can be seriously disturbed by delays, accidents or technical problems. Major disturbances cause primary delays that propagate as consecutive delays to other trains in the network, thus requiring short-term adjustments to the timetable in order to limit delay propagation. This real-time problem is known as Conflict Detection and Resolution (CDR).

Keeping transfer connections when solving the CDR problem increases delay propagation (Vromans, 2005), therefore one of the possible dispatching countermeasures to handle disturbances is the cancellation of some scheduled connections. This action reduces the overall train delays but has a negative impact on passenger satisfaction for the transferring passengers affected by missed connections. Train operating companies are therefore interested in keeping as many connections as possible even in the presence of disturbed traffic conditions, while infrastructure managers are mainly interested in limiting train delays. In fact, infrastructure managers discuss with train operating companies on which connections must be kept when regulating railway traffic. To support this negotiation process, this paper deals with a Bi-objective Conflict Detection and Resolution (BCDR) problem (Corman et al, 2010), i.e., the problem of finding a set of feasible schedules with a good trade-off between the minimization of train delays and the maximization of respected transfer connections.

The BCDR problem is closely related to the Delay Management (DM) problem introduced by Schöbel (2001). The latter problem adopts a passenger point of view, and aims at the minimization of the sum of all delays over all passengers at their final destination. In this paper we choose a train point of view, i.e., the minimization of train delays at all relevant

points and the maximization of the total value of respected connections. A value is associated to each connection between trains (e.g., expressed in terms of the number of passengers who get the connection, even though the model does not depend on the criteria used to define the relevance of a connection). A further difference is that the DM problem does not take into account the limited capacity of the railway network and, in fact, does not deal with the CDR problem. Neglecting this constraint may result in timetable adjustments that cause conflicts when put into operations; this is prevented when an approach featuring a precise modeling of train movements is used, as the one proposed.

We model the BCDR problem as a special bi-objective job shop scheduling problem, which is formulated by alternative graphs (Mascis and Pacciarelli, 2002). A node $i \in$ N of the alternative graph is associated to the starting time $t_i$ of the $i$-th relevant event. The set F of fixed arcs is used to represent precedence between events. The set A of alternative arcs is used to represent train sequencing decisions. A further set C of connection arcs represents connections enforcement. Our formulation is next represented by the disjunctive program, i.e., a linear program with logical conditions involving operation "or" ($\vee$, disjunction):

$$\begin{array}{lll}
\min \{t_n - t_0; -\sum_{(i,j)\in \mathrm{C}}[v_{ij}\, \delta(t_j - i_i - w_{ij})]\} & & \\
\text{s.t.} & t_j - t_i \geq w_{ij} & (i, j) \in \mathrm{F} \\
 & (t_j - t_i \geq w_{ij}) \vee (-) & (i, j) \in \mathrm{C} \\
 & (t_j - t_{\sigma(i)} \geq w_{\sigma(i)j}) \vee (t_i - t_{\sigma(j)} \geq w_{\sigma(j)i}) & ((\sigma(i)j),(\sigma(j),i)) \in \mathrm{A}
\end{array}$$

where σ(i) is the operation which follows $i$ on the route of the associated train, and the precedence relation $t_j - t_i \geq w_{ij}$ ensures that $j$ starts after $t_i$ plus a time lag $w_{ij}$, as described in (D'Ariano et al., 2007); (−) represents a dummy constraint. The two objective functions are: the minimization of the maximum consecutive delay $t_n$-$t_0$ and the maximization of the total value of all connections enforced. The function $\delta(x)$ is equal to 1 if $x \geq 0$ and is equal to 0 if $x < 0$. $\delta(x)$ is used to take into account the value $v_{ij}$ of each connection kept (i.e., those connections for which the time lag is respected and $t_j \geq t_i + w_{ij}$).

The solution procedure consists of estimating the Pareto front of non-dominated solutions for the BCDR problem. The solution strategy adopted in this paper consists of iteratively solving the CDR problem (with fixed connections) and then searching for a different set of connections to be enforced. For the selection of the connections to be enforced, we develop and test two new algorithms for the BCDR problem, called Add and Remove, based on the meta-heuristic framework proposed by Paquete and Stützle (2006). Both algorithms use the branch and bound described in (D'Ariano et al., 2007) to solve the CDR problem and maintain an archive Z of Pareto-optimal solutions which is returned at the end of the search.

In what follows, C is the set of all transfer connections to keep or drop and $S^C \subseteq C$ is the set of enforced transfer connections. We let $D(S^C)$ be the maximum consecutive delay associated to an optimal solution to the CDR problem for a given set $S^C$. We also let $V(S^C)$ be the total value of the connections satisfied in this solution. The pair $[V(S^C), D(S^C)]$ is the associated point in the plane of the two objective functions for the BCDR problem. Each solution in the archive is characterized by the set $S^C$ with attributes $[V(S^C), D(S^C)]$ and a visited flag $f(S^C)$ initially set to 0. This flag is used during the search to keep track of the already visited solutions (with $f(S^C) = 1$). Initially, a starting solution is inserted in the archive depending on the chosen algorithm ($S^C = \varnothing$ for the Add algorithm and $S^C = C$ for the Remove algorithm). A neighbor $\hat{S}^C$ is the set obtained by adding to $S^C$ a single connection in $C - S^C$ (algorithm Add) or removing a single connection from $S^C$ (algorithm Remove).

We now show the sketch of algorithm Add.

```
Set $S^C = \emptyset$, compute $V(S^C)$ and $D(S^C)$
Initialize archive Z with $S^C=\emptyset$ with attributes $[V(S^C), D(S^C)]$ and visited flag $f(S^C)=0$
    while there is at least an element in the archive with $f(S^C) = 0$ do
        Select an element $S^C$ with $f(S^C) = 0$ from the archive Z
        for all connections $j \in C - S^C$ do
            Generate a neighbor $\hat{S}^C = S^C \cup \{j\}$
            if the set $\hat{S}^C$ is not in Z do
                Compute $V(\hat{S}^C)$ and $D(\hat{S}^C)$
                Append $\hat{S}^C$ to Z with $[V(\hat{S}^C), D(\hat{S}^C)]$ and $f(\hat{S}^C) = 0$
            end if
        end for
        Set $f(S^C) = 1$
        Remove from Z all the dominated elements
    end while
```

The computational study is based on the railway network around the main station of Utrecht, in the Netherlands. We use a peak hour of the 2008 timetable with 80 trains, mostly passenger trains and a few freight trains, and 451 resources, either block sections or platforms. The resulting alternative graph has $|N| = 1847$ nodes, $|F| = 2156$ arcs, $|A| = 4773$ pairs of alternative arcs. As for the set C of connections, we analyze three scenarios. The first one includes 12 passenger connections and 7 non-relaxable rolling stock connections, as in real operations. The second, more challenging, scenario includes 24 relaxable connections. A third scenario is generated by modifying this latter connection set with artificially longer connection times, which result in delay propagation even in absence of entrance train delays. For each scenario, a set of 25 perturbation instances is generated for different values of entrance train delays. Trains are delayed at their entrance into the network up to a maximum of 1313 seconds, with an average entrance delay of 181.6 seconds. About 25% of all trains have an entrance delay of more than 5 minutes.

Table 1 describes the impact of keeping connections for the BCDR problem. The gap is computed between the two extreme cases in which all relaxable connections are dropped (Columns 1-2) or enforced (Columns 3-4), for the 25 perturbations and the three scenarios, and shown in terms of maximum consecutive delay (in seconds) and the total value of the relaxable connections satisfied (also when not enforced). The large time reserves in the timetable make most of the connections satisfied automatically in the first two scenarios, even though keeping all connections increases significantly the propagation of delays compared to the case in which a few connections are not satisfied. The artificial third scenario has much larger connection times, resulting in fewer connections satisfied even if not enforced, and larger delay propagation.

**Table 1: Gap between the values of the extreme BCDR solutions**

| *No Connection Enforced* | | *All Connections enforced* | |
|---|---|---|---|
| *Max Consecutive Delay (s)* | *Connections Value* | *Max Consecutive Delay (s)* | *Connections Value* |
| First Scenario | | | |
| 208.7 | 12.8 | 530.1 | 16 |
| Second Scenario | | | |
| 208.7 | 69.6 | 583.6 | 76 |
| Third Scenario | | | |
| 208.7 | 28.3 | 1265.2 | 76 |

Table 2 reports on the Pareto front of non-dominated solutions generated by the Add and Remove algorithms. For the first scenario, we also compare the performance of the two algorithms with the Pareto front computed by enumerating all possible combinations of enforced connections. For each algorithm, Column 2 shows on the average number of instances of the CDR problem that have to be solved for an instance of the BCDR problem, Column 3 the average time required to compute the Pareto front (in seconds) and Column 4 the number of Pareto-optimal solutions. Then, Column 5 gives the Pareto front area, i.e., the ratio between the area covered by the Pareto points found in the solution space and the area resulting by drawing the rectangle of the two extreme non–dominated points (one point is obtained by enforcing all connections and the other point is obtained by no connection enforcement). Column 6 presents the percentage of times the branch and bound code reaches the time limit of computation before proving the optimality of its current best CDR solution.

**Table 2: Average results on the Pareto front algorithms**

| *BCDR Algorithm* | *Scheduler Calls* | *Total Time (s)* | *# of P.O. solutions* | *P.F. Area (%)* | *Time Limits BB (%)* |
|---|---|---|---|---|---|
| First Scenario | | | | | |
| Remove | 36 | 283 | 3.32 | 26 | 18 |
| Add | 20 | 166 | 3.32 | 26 | 23 |
| Exhaustive | 4096 | 33504 | 3.32 | 26 | 20 |
| Second Scenario | | | | | |
| Remove | 91 | 705 | 4.04 | 28 | 18 |
| Add | 36 | 309 | 4.00 | 27 | 25 |
| Third Scenario | | | | | |
| Remove | 229 | 1867 | 12.96 | 60 | 18 |
| Add | 149 | 1123 | 14.08 | 63 | 24 |

Both algorithms are very effective in generating the Pareto front. For the first scenario, Add and Remove find the same Pareto front of the exhaustive search within less than 1% of computation time. For the second scenario, the number of Pareto-optimal solutions found by Add algorithm is only 1% less than Remove. Algorithm Add is quite more efficient than algorithm Remove, its computation time being only 58% [respectively 44%] than the computation time of Remove for the first [the second] scenario. Concerning the third scenario, Add outperforms Remove for all the performance indicators, whereas the area covered is significantly larger than for the first two scenarios, as well as the number of solutions found on the Pareto front and the total computation time, due to the artificial connections considered.

From the experiments, carried out on real-world data, we conclude that good coordination of connected train services is important to achieve real-time efficiency of railway services since the management of connections may have a serious impact on delay propagation. Both algorithms are very effective in approximating the Pareto front within a limited time of computation.

Overall, these results demonstrate that finding a compromise solution between delay minimization and connection satisfaction deserves a high potential for advanced performance management and for the development of a real-time decision support system. Future research should be dedicated to include additional needs of the different stakeholders, other practical constraints, dispatching measures and performance indicators. Another interesting research direction is the extension of the BCDR problem to other transportation domains like air traffic management.

## REFERENCES

Corman, F., A. D'Ariano, D. Pacciarelli, M. Pranzo (2010) Bi-objective conflict detection and resolution in railway traffic management, in: *Transportartation Research, Part C*, in press.

D'Ariano, A., D. Pacciarelli, M. Pranzo (2007) A branch and bound algorithm for scheduling trains in a railway network, in: *European Journal of Operational Research* 183 (2), pp. 643–657.

Mascis, A., D. Pacciarelli (2002) Job shop scheduling with blocking and no-wait constraints, in: *European Journal of Operational Research* 143 (3), pp. 498–517.

Paquete L., T. Stützle (2006) A study of stochastic local search algorithms for the bi-objective QAP with correlated flow matrices, in: *European Journal of Operational Research* 169 (3), pp. 943–959.

Schöbel, A. (2001) A model for the delay management problem based on mixed integer programming, in: *Electronic Notes in Theoretical Computer Science* 50 (1), pp. 1-10.

Vromans, M.J.C.M. (2005) *Reliability of Railway Systems,* TRAIL Thesis Series T2005/7, the Netherlands.