Automated dispatching of train operations using genetic algorithms

S. Wegele & E. Schnieder
Technical University Braunschweig,
Institute for Traffic Safety and Automation Engineering,
Langer Kamp 8, 38106 Braunschweig, Germany

Abstract

Considering the German transportation growth prognosis (Verkehrsprognose für die Bundesverkehrswegeplanung) for the year 2015 the rail traffic performance has to improve drastically in order to avoid possible traffic chaos on German roads. According to this it is especially necessary to encourage the competitiveness of rail traffic. Therefore the advantages of road traffic - flexibility and punctuality - have to be established comparably in rail traffic. Presently the railway management is based on a centralized supervision and dispatching of railway operation, which has to be optimized in order to handle the upcoming growth of transportation. This task can be performed by automated train path routing procedures based on mathematical optimizations. This paper shows a successful promising approach using genetic algorithms. The application of this stochastic optimization procedure requires a highly efficient simulation model of railway operations, which in the present case has been implemented by using Petri-nets. On the bases of a practical case study the possible fields of application for this prototypical implemented procedure of train path routing are presented.

1 Motivation

The continuous economical growth in European countries during the last twenty years is closely connected with a nearly exponential increase of traffic volumes on road and rail. The forecast for the next 15 years is an expected near 50% increase of road traffic volume in Germany which, despite an improvement in road infrastructure, could lead to a collapse. One way to deal with this threatening forecast is to increase the transport capacity of rail transport. To achieve this aim in challeng-

Computers in Railways IX, J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto & S. Sone (Editors)
© 2004 WIT Press, www.witpress.com, ISBN 1-85312-715-9

ing free market conditions, the punctuality and flexibility of transport by rail must be comparable to that by road.

These goals are only achievable with very fast train traffic planning (flexibility) and optimal online dispatching (punctuality). At present, the main part of planning tasks is carried out by humans. With an increase in transport volume and decrease in available planning time, the complexity of decision making increases exponentially and is therefore no longer so manageable by human planners. Computer based support of this task is essential for rail transport companies.

To deal with this task many mathematical approaches have been investigated during the last three decades [1, 2, 3, 4, 5]. Their common drawbacks are either too low precision or too high computational efforts. This is based on the fact, that the planning task is NP-complete and there is no deterministic method, which can find the solution with polynomial complexity.

2 Approach

The actual planning process carried out by railway company management can be modelled as a control task consisting of two iterative control loops (fig. 1).

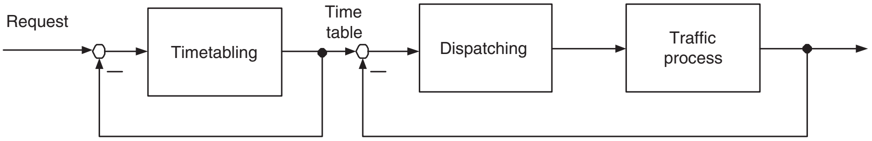

Figure 1: Control loops in transportation process.

In the frame of the first control loop the trains are scheduled according to the requirements of transport providers. This is an iterative process, with the aim of finding out the best compromise between available slots and customer requirements. Several factors must be considered: besides source-destination relationships, additional factors like prices, priorities, required punctuality etc are very important. All these factors have to be implemented on train paths and time instants in the form of a time table. This provides an input for the second control loop dealing with operative dispatching.

The second control loop is responsible for online train control reacting on actual disturbances in the transport process. This often requires re-scheduling of the original time table, in order to minimise the consequences of operational conflicts. The main difference is that the first control loop represents a much shorter reaction time. While train planning can be performed within several hours or days, dispatching has to be effected within minutes or even seconds.

Computers in Railways IX, J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto & S. Sone (Editors)

2.1 Formal definition of the optimisation goal

The goal of an optimisation method can be formulated by a function:

$$f(x_1, x_2, x_3, \cdots) \rightarrow \min \qquad (1)$$

The optimisation method aims to minimize the value of the function by variation of the function parameter x_i; in connection with a timetable this would lead to the best fulfilment of customer's requirements. In the form of a function:

$$\{\text{Deviation from requirements}\} \rightarrow \min$$

These requirements vary during the planning process:

- At the beginning: time intervals for departure and arrivals, source and destination points, prices for track occupancy.
- After providing the time table: possibly the smallest deviation of train operations from the time table.

For both kinds of requirements the following penalty function can be formulated:

$$\sum f_s(t_{\text{is}} - t_{\text{should}}) + \sum f_s(V_{\text{is}} - V_{\text{should}}) \rightarrow \min \qquad (2)$$

with

f_s : penalty function.
t_{is} : actual arrival of the trains.
t_{should} : desired arrival of the train.
$V_{\text{is}}, V_{\text{should}}$: actual and desired train connections.

During minimisation of the resulting penalty function 2 the following aspects must be taken into account:

- It is not always possible to fulfil customer requirements (the penalty function results in $F > 0$). The minimal value of F is mostly unknown.
- The single train paths (slots) depend mutually on each other. Because of sharing the same railway infrastructure resources, they compete for the limited traffic throughput.

Because of that the goal of the optimisation method is to find the best compromise among all customers' requirements and the given constraints. In order to build a relationship between single requirements, the corresponding penalty functions must contain the weighting factors e.g. an ICE has higher penalties for delay then a freight train.

2.2 Dispatching method

An optimisation based dispatching can be seen as a cyclic improvement process (s. fig. 2). The optimisation method creates new time tables according to actual disturbances by using the variation and combination of the already calculated solutions. After that an evaluation of the created solutions follows. This part of the algorithm consumes more than 80% of the compute time. That was the reason for developing the effective discrete-event simulation based on Petri net (s. 2.5).

Computers in Railways IX, J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto & S. Sone (Editors)
© 2004 WIT Press, www.witpress.com, ISBN 1-85312-715-9

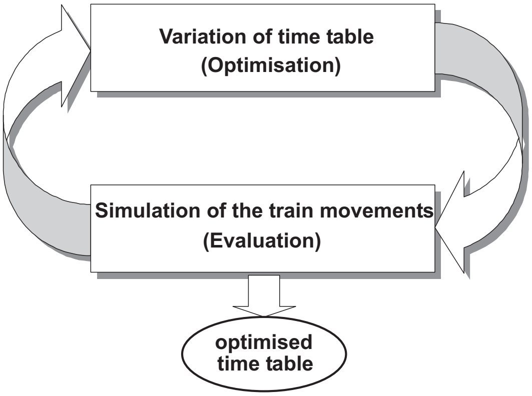

Figure 2: Optimisation process.

2.3 Optimisation methods

The optimisation task for dispatching can be mapped on discrete and continuous variables. Accordingly, this task can be solved by different optimisation methods (s. fig. 3).

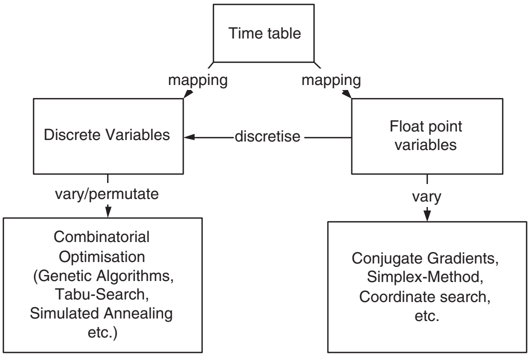

Figure 3: Alternatives for coding of the time tables.

The suitable coding of the optimisation problem allows the use of more efficient methods from the branch of combinatorial optimisation [6].

Combinatorial optimisation methods can be divided into two groups: constructing (e.g. Branch and Bound) and (iterative) improving (e.g. Genetic Algorithms) methods. Constructing methods can be seen as a search in a decision tree, where the whole solution is ”constructed” only at the end of the search, when all decisions have been made. The improving methods need a whole solution before they can start optimising. In our work we combined both methods:

- Branch-and-bound method was used to construct the first solution taking into account only a restricted amount of possible decisions. Because of the restriction made, this optimisation method could find a good start point in appropriate computational time, despite the exponential complexity of the method.
- As an iterative improving method we applied the genetic algorithms. Their advantage is that they can be combined with many other optimisation techniques like Tabu-search, simulated annealing etc.

The genetic algorithm is based on a model of evolution by John Holland [7]. It models the following factors of Nature:

- There is a compact coding defining individuals. In Nature this task is carried out by chromosomes. In our case they code the time table. The coding is based on train chains connected to nodes of the topology graph.
- There is a rule about how the coding can be transformed to an individual. In our case it is a mapping of the time data from the time table to real train movements on the railways.
- There are sexual as well asexual reproductions; i.e. there is a possibility of combining the features of single individuals (cross-over). In our case some genetic operators are defined by combining parts of the different time tables e.g. taking one train from one timetable and another from the other one.
- There is a mechanism to evaluate the fitness of individuals (natural selection)). In our case it is a penalty function 2.
- There is a stochastic influence on the coding: mutation. In our case it is a stochastic conflict based manipulation of the coded train chains.

2.4 Reduction of the complexity for searching

Even with an appropriate coding of the time table in the form of train-chains, the resulting searching space is too big to be optimised by a stochastic optimisation method. To reduce the calculation amounts, an additional level of coding of the optimisation problem is proposed. In order to localise the possibility with a high improvement potential, the train interaction have to be analysed. For this purpose a conflict tree has been constructed (s. fig. 4).

It allows for the recognition of independent conflict chains with trains involved in the conflict. Their representation is based on a sequence diagram of Unified Modelling Language (UML) [8, 9]. The white blocks stand for the track occupation time by the trains and the black arrows mean the delay time transferring from one train to the other. Their width codes are the quantitative amount of the corresponding delay. In fig. 4 a conflict chain $ICE768 \rightarrow IC6574 \rightarrow RB76438 \rightarrow RB67854$ is shown as an example. This conflict chain involves only a small number of variables and that leads to an effective reduction of the searching complexity. Another benefit of the coding lies in the possibility of an independent solution of such conflict chains on multiple computers hence the distribution of the calculations. The resulting data flow of the algorithm is shown in fig. 5.

Even with the introduced coding of the optimisation problem the rest searching

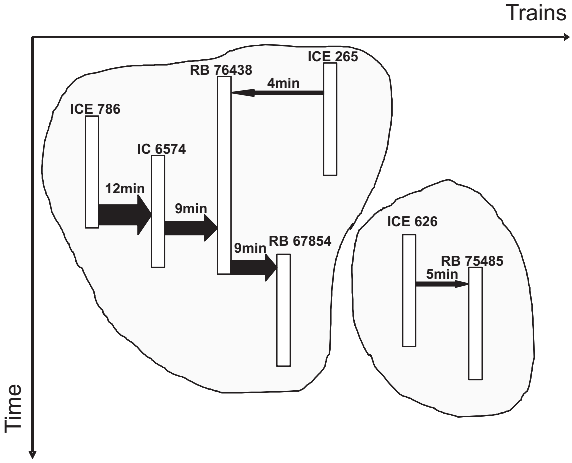

Figure 4: Conflict diagram.

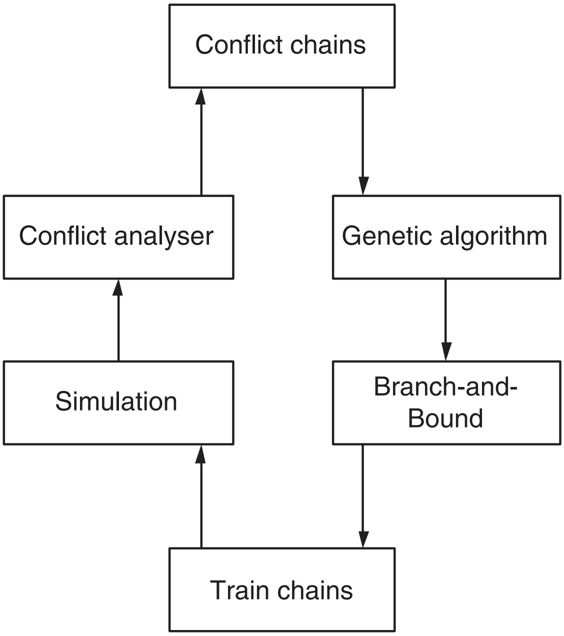

Figure 5: Data flow in the optimisation cycle.

space can still be very big to be analysed within a desired time interval. In almost all the practical tasks the quality of solution depends on the number of analysed alternatives. As the analysis of the conflict chains and proposal of possible solutions is a relatively simple task, the remaining calculation effort (about 80%) is consumed by the simulation of train movements. In the next section we show an effective discrete-event simulation model based on Petri net [10].

2.5 Simulation

The simulation of the train movement consists of calculating train arrivals and departures on some points of the infrastructure. So the simulation depends directly on the number of topology elements considered. One way to reduce the simulation efforts is to reduce the number of considered elements as much as possible. This can be done by the grouping of elements and handling the simulation with such groups. This kind of grouping takes place by combining the elements of the infrastructure into blocks. Blocks are bounded by signals and consist of all the elements between them. To avoid train collisions the blocks must be used by one train only. Block edges can be easily modelled to be a transition of a Petri net (s. fig. 6).

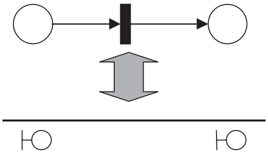

Figure 6: Mapping of the rail block to a Petri net.

The tracks before signals are represented by places and the railway line between them by transition. After such a transformation of the infrastructure to Petri net by this rule the result cannot be used for simulation yet. The reason is that every element of the infrastructure can be used in at least two blocks: one in each direction. If the blocks are sharing some elements the corresponding transitions must be coordinated by a common place. One possibility to achieve the coordination is to combine two signals in a different direction to one place of Petri net (s. fig. 7). This is only possible if there are no switches between these signals. Otherwise a switch must be modelled by an additional place. The fact that every block can be used by only one train is automatically taken into account: the Petri net defines the switch rules, where one transition is allowed to fire, only if the following place is empty.

3 Validation on a practical application

The decision support system introduced here was implemented at the Institute for Traffic Safety and Automation Engineering of Technical University Braunschweig. It was validated on practical data delivered by Deutsche Bahn AG (fig. 8). A relatively big sector of the German railways net in Northern Germany (1% of topology and 2% of the traffic volume) has been transformed to a Petri net. It incorporates

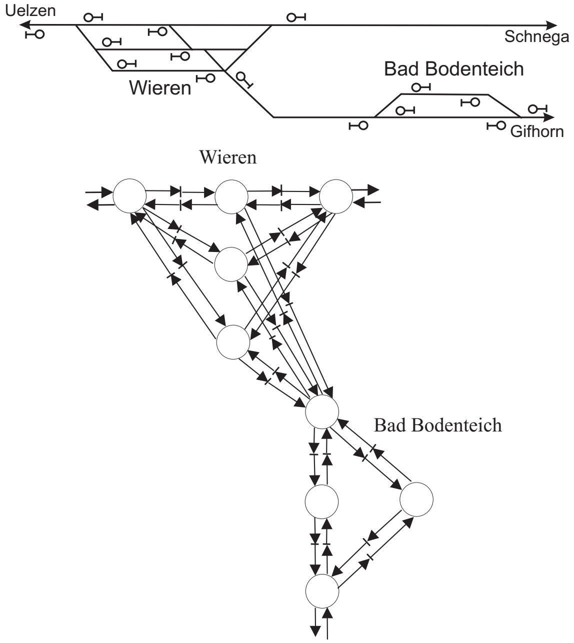

Figure 7: Mapping of railway infrastructure into Petri net.

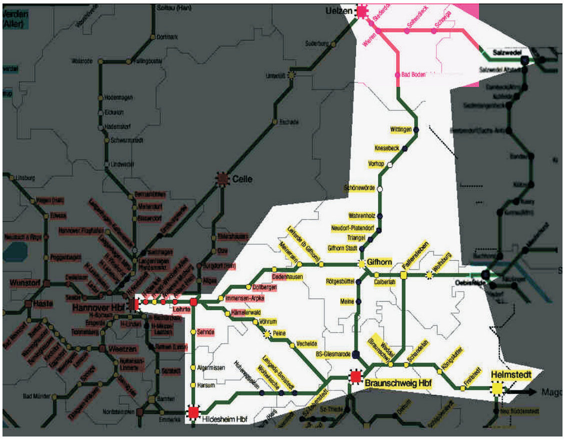

Figure 8: Considered part of German railways.

single line as well double line tracks and consists of 104 stations. More than 1000 passenger and freight trains operate during 24 hours.

The considered benchmarks contain the following use cases:

Computers in Railways IX, J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto & S. Sone (Editors)
© 2004 WIT Press, www.witpress.com, ISBN 1-85312-715-9

- Inserting a new train line with the certain priority and arrival times into a filled time table.
- Replanning of all trains because of stochastic disturbances in the entering time of the trains into the considered area. The disturbances were modelled by a stochastic value with uniform distribution inside of [0:30] minutes. One optimisation run is shown in fig. 9. As a reference for the analysis of the optimisation quality we used a theoretical situation, where every train has its own rail track without interference with any other train. The theoretical improvement in % was calculated by the formula:

$$R = 1 - \frac{(P_{result} - P_{theoretical})}{P_{result}} \tag{3}$$

R : optimisation residue.
P : penalty for particular time table.

After three minutes of evaluation the remaining optimisation potential lay at 11.2% (s. fig. 9).

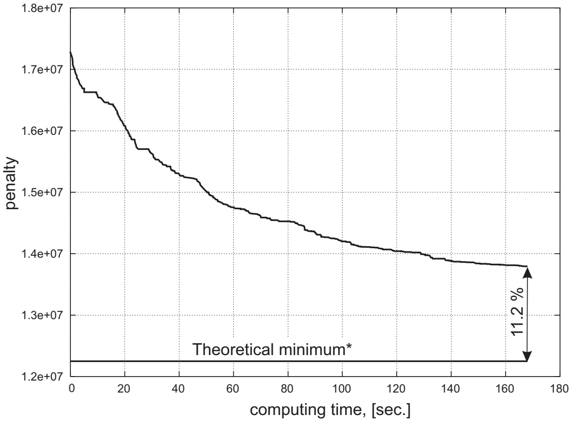

Figure 9: Optimisation run for stochastically disturbed railway traffic.

The evaluation runtimes in both cases lay at a few minutes, which is enough for practical application of the method.

Acknowledgement

We are grateful to the Deutsche Bahn AG for sponsoring and support of this work.

References

[1] Kraft, E.R., A branch and bound procedure for optimal train dispatching. *Journal of the Transportation Research Forum*, **28**, pp. 263–276, 1987.

[2] Kraay, D., Harker, P. & Chen, B., Optimal pacing of trains in freight railroads: Model formulation and solution. *Operations Research*, **39**, pp. 82–99, 1991.

[3] Jovanovic, D. & Harker, P.T., Tactical scheduling of rail operations: The scan i system. *Transportation Science*, **25(1)**, pp. 46–64, 1991.

[4] Brnnlund, U., Lindberg, P.O., Nu, A. & Nilsson, J.E., Railway Timetabling using Lagrangian Relaxation. *Transportation Science*, **32(4)**, pp. 358–369, 1998.

[5] Blum, J. & Eskandarian, A., Domain-specific genetic agents for flow optimization of freight railroad traffic. *Computers in railways VIII*, WIT Press, pp. 787–796, 2002.

[6] Reeves, C.R., (ed.) *Modern heuristic techniques for combinatorial problems*. McGraw-Hill, 1995.

[7] Holland, J., *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[8] Fowler, M., *UML Distilled*. Addison-Wesley, 2003.

[9] OMG, *Unified Modeling Language Specification*. Object Management Group, 1999.

[10] Petri, C.A., *Kommunikation mit Automaten*. Ph.D. thesis, TH Darmstadt, 1962.

Computers in Railways IX, J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto & S. Sone (Editors)