KRONECKER ALGEBRA FOR OPTIMIZATION OF RAIL TRAFFIC FLOW ON ZAGREB – RIJEKA LINE

Andreas SCHÖBEL\textsuperscript{1}
Christian SCHÖBEL\textsuperscript{1}
Stefan MARK\textsuperscript{2}
Hans Blieberger\textsuperscript{3}

Abstract – Within the Shift2Rail-Project "GoSafeRail" so called Kronecker Algebra is applied on the railway line from Zagreb to Rijeka for traffic flow optimization. Kronecker Algebra consists out of Kronecker Product and Kronecker Sum to describe concurrency of tasks as well as their inter-leavings. The railway line from Zagreb to Rijeka has been already modeled by software tool called OpenTrack. In this project data from OpenTrack model is automatically converted into input files for a 3rd party tool based on Kronecker Algebra. To model the infrastructure so called “IVT” format is used to export an itinerary covering the main track. The entire line consists out of 250 edges while station areas with more than one station track are handled by counting semaphores. The timetable is exported in OpenTrack text format and used to create train run files. On a normal working day around 115 trains are running on this line but of course not all trains use the entire line. Additionally, train characteristics are considered to calculate realistic running times.

Keywords – Railway operation, rail traffic flow, optimization

1. INTRODUCTION

One of the objectives of the Shift2Rail project Global SAFety Management Framework for RAIL Operations is the development of an evolutionary Decision Support Tool that self-learns (evolves) based on machine learning algorithms and artificial intelligence with the main goal of offering safer, reliable and efficient rail infrastructure. Due to a low number in failures on the infrastructure network, this leads to a lack of data crucial for machine learning. This will be solved by implementation of Near-Miss Concept; in other words, low-consequence events will be also included in the model and enable use of statistically significant data for model training. Furthermore, a new train mounted multiple sensor system for Object Detection will be developed.

Moreover, with OpenTrack \cite{opentrack} micro-simulation modelling tool, traffic model will be developed that will use multi-criteria optimization algorithms to address complex requirements, for both passenger and freight transport. Using Kronecker algebra, which showed good results in dealing with optimization scenarios in railway traffic flow, especially avoidance of deadlocks, simulation of the line between Zagreb and Rijeka in Croatia has been performed as well as a optimisation based on Kronecker Algebra.

2. KRONECKER ALGEBRA FOR RAILWAY OPERATION

One of the constantly present problems in railway systems is the problem with deadlocks especially on single track lines or during rehabilitation work \cite{rehabilitation} on a double track line and around dead end hubs. Since there were no applicable solutions in the middle of 20th century, computer scientists tried to solve this problem by implementing Kronecker algebra in the analysis \cite{kronecker}.

Before going into solving deadlock issue, a proper definition is needed. Stallings \cite{stallings} defines Deadlock as ‘an impasse that occurs when multiple processes are waiting for the availability of a resource that will not become available because it is being held by another process that is in a similar wait state’. There are four preconditions for a deadlock to occur according to Coffman \cite{coffman}; in other words, if one of these conditions is not met, there will not be a deadlock. There is a mutual exclusion, where a resource can only be used by one process at a time. Second, hold and wait includes processes already holding resources and

\begin{thebibliography}{9}
\bibitem{opentrack} OpenTrack Railway Technology GmbH, Kaasgrabengasse 19/8, A-1190 Wien, andreas.schoebel@opentrack.at
\bibitem{coffman} Austrian Institute of Technology, Giefinggasse 2, A-1210 Wien, mark.stefan@ait.ac.at
\bibitem{kronecker} Vienna University of Technology, Treitlstraße 1-3, A-1040 Wien, blieb@auto.tuwien.ac.at
\end{thebibliography}
requiring additional resources held by other processes. Third, the so-called no preemption, no other than the process itself can release the resource. Finally, the circular wait that requires at least two processes to form a circular chain in which each process waits for a resource that is being held by the previous process in the chain. Clearly, these four conditions can be applied to railway systems.

After defining conditions for deadlock occurrence, possible ways to deal with deadlocks can be identified. These are deadlock prevention, or removing one of the above mentioned conditions in order to prevent deadlock from even occurring; deadlock avoidance, or decision about resource allocation in advance; and finally, after deadlock detection, termination and restart of the process. For the railway systems, only deadlock avoidance is applicable [3].

Kronecker algebra is a mathematical model that consists of Kronecker Sum and Kronecker Product. For the explanation of these two operations, set of matrices (1) is defined

\[ M = \{ M = (m_{ij}) | m_{ij} \in L \} \quad (1) \]

where \( L \) denotes a set of labels with \( (L,+,0) \) being a commutative monoid and \( (L,\ast,0) \) a monoid of order \( mn \) defined by

\[ A \oplus B = A \otimes I_n + I_m \otimes B \quad (4) \]

where \( I_m \) and \( I_n \) (n-by-n matrix with ones on the main diagonal and zeros elsewhere) denote identity matrices of order \( m \) and \( r \), respectively.

Application of Kronecker algebra in optimization of railway traffic flow lies in its functionality to detect and avoid any deadlocks within the whole analysed railway system, not just on one section. To put it differently, it can is represented as a matrix that includes all possible train movements in a system. In other words, deadlock-free solutions are overall best calculated solutions that take schedules, delays and different types of restriction on the tracks into account [7]. Whereas Kronecker Sum calculates all possible interleavings of all trains not using the same track section, Kronecker Product ensures that those using common track sections can sequentially enter only free sections, namely, sections previously released by another train. Kronecker Algebra delivers results as a matrix. However, these can be represented as a graph, especially time-speed diagram.

3. USE CASE OF ZAGREB – RIJEKA LINE

Within GoSafeRail project [2] the railway line from Zagreb to Rijeka has been selected. As a first approach, the edges for the Kronecker algorithm were created between signals, speed changes and gradient changes. With this ruleset, the resulting number of edges was 1067 for the Zagreb Rijeka line, which was a fairly high amount for the Kronecker algorithm. In order to reduce execution time of the Kronecker algorithm, the infrastructure was partitioned in a macroscopic view (tracks.csv) and a microscopic view (tracks-micro.csv). For the macroscopic view, edges where only created from signal to signal. With this measure, the number of edges for the Zagreb Rijeka line was cut down to 250. Figure 1 depicts the result of this reduction, reimported into OpenTrack via IVT input.

Another major aim in building the infrastructure for Kronecker was the simplest possible...
representation of the stations. The o2k converter takes care of this by producing only one edge per station, which receives the number of available tracks at this station as the maximum value for its semaphore.

After creating the infrastructure, the second step includes the production of all train course data files required for Kronecker operation by the o2k converter. OpenTrack’s timetable data gives information about the passing location of the individual train. If corresponding data is available, time passes from the timetable are used as so-called measure points for Kronecker optimization. Train course files for Kronecker must reserve and free edges in a proper sequence. Especially at stations where a train should stop, the pattern for reserving and freeing edges seems to be somewhat complicated. The screenshot shown in figure 2 of a train course file generated by the o2k converter is presented in order to clarify what the proper solution to this problem shall look like.

The lines starting with a hash mark (#) are comments and will be ignored by the input parser of the Kronecker implementation. These comments have proved to be useful for finding errors contained in the original data from OpenTrack, which has been entered by manually and thus may be erroneous.

For this explanation, lines will always be called by the number at their beginning. Line 172 shows a real stop taking place. Real stops contain the code 10 or 20 at the third field, depending on this course’s direction: 10 is for the up direction, 20 is for the down direction. The direction is set by the itinerary of the original IVT file from OpenTrack.

The last column of each line contains a P or V operation for a given edge of the Kronecker infrastructure, which can be found in the tracks.csv file. Following common naming conventions, P is used for reserving an edge and V is used for freeing an edge.

At line 172 in this example, a stop takes place when reserving edge 163. Consulting the generated infrastructure file tracks.csv for the Zagreb Rijeka line, it can be seen that edge 163 is the station area of Duga-Resa with a maximum semaphore value of 3, meaning that there are three tracks in Duga-Resa which can be used for halting trains or overtaking stopped trains.

Line 173 shows the next operation which takes place after the train has stopped at Duga-Resa. This operation is of high importance for Kronecker algorithm. The train arrives via the edges 167, 166, 165, 164 until it halts at edge 163. The edges 167, 166 and 165 are freed in the lines 169 to 171 as can be seen in the sample snipped above. This always happens as a preparation prior to a stop and is called a “flush” in terms of the o2k converter. The edge 164 must not be freed before the train stops at edge 163, but must be freed immediately after. The o2k converter takes care of this.

When a train arrives at its final station, the codes 100 or 200 are used for the third column. As mentioned above, prior to a real stop, a flush is conducted for all but the last edges. Similarly, the last edge before the stop is freed immediately after the stop. The station edge, where the real stop itself is performed, will be freed five minutes after the stop has occurred.

<table>
<thead>
<tr>
<th></th>
<th>Earliness in OpenTrack</th>
<th>Earliness in Kronecker</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sum</strong></td>
<td>-3143</td>
<td>-2779</td>
</tr>
<tr>
<td><strong>Number of Trains</strong></td>
<td>42</td>
<td>35</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>-75</td>
<td>-79</td>
</tr>
</tbody>
</table>

Tab. 1. Overview on earliness in seconds calculated by Kronecker and simulated in OpenTrack
Table 1 and 2 compare earliness and delays from simulation in OpenTrack of all 57 daily passenger trains using the entire or parts of the Zagreb-Rijeka line with calculated values by Kronecker. While trains arrive earlier in sum in OpenTrack simulation, delays are reduced by application of Kronecker in sum and in average. Further investigations will be carried out on the level of each single train run to validate the calculations of Kronecker by simulations of OpenTrack. Thereby, the recommendations from Kronecker can be used as input for the actual performance of each single train.

4. CONCLUSION

OpenTrack, being a sophisticated micro-simulation model allows the determination of impact of safety decisions on operational network performance. Thus, by incorporating both infrastructure asset (e.g. crossings, tracks, bridges, tunnels) and traffic (e.g. vehicle, freight and passenger movement), effective delivery of maintenance or new works while maximising the connectivity and adaptability of the overall surface system will be enabled. As a performance indicator for Kronecker Algebra the delays of trains at their final station are used as a benchmark criteria. Latest test runs on the railway line from Zagreb to Rijeka show a reduction of delay of more than 60 % in sum and 70 % in average.

ACKNOWLEDGEMENT

GoSafeRail project has received funding from Shift2Rail under grant agreement No 730817.

REFERENCES