NESTED SIMULATIONS SUPPORTING TRAFFIC OPTIMIZATIONS RELATED TO RAILWAY STATIONS

Roman Diviš[(a)], Antonín Kavička[(b)]

[(a),(b)]University of Pardubice, Faculty of Electrical Engineering and Informatics

[(a)]roman.divis@upce.cz, [(b)]antonin.kavicka@upce.cz

ABSTRACT

Nested simulations present a general method suitable for use in realizing a multi-trajectory simulation or as a decision support in a simulator. The principle of nested simulation (as a decision support) is to find a solution to a problem using other time-limited simulations which verify alternative options. After the nested simulations have finished, the solutions of individual alternatives are assessed and the best solution is applied to the main simulation.

The aim of the article is to conduct a case study of using nested simulation as decision support in a mesoscopic simulator of rail transport. This study is based on a prototype railway station with mixed traffic of both passenger and cargo trains. The results of the simulation are compared with commonly used microscopic simulation tool Villon.

Keywords: nested simulations, decision support, rail transport

1. INTRODUCTION

The scope of examination is simulation of rail transport. At the moment, the focus is placed on realisation of own mesoscopic simulation tool called MesoRail (Diviš and Kavička 2015). The aim of the simulator is to allow the examination of both the deterministic and the stochastic operation, mainly within the scope of railway stations. Stochastic simulators deal with random events within the simulation, during which conflict situations, with possibly not known optimal solutions within defined rules, occur. When such events occur, it is possible to use a decision support subsystem, aim of which is to provide information for solving particular conflict states. There is a variety of methods and approaches suitable for realizing decision support. This article further focuses on using the method of nested simulations as decision support.

2. THE METHOD OF NESTED SIMULATIONS

The method of *nested/recursive simulations* presents a principle of using simulation inside a simulation in order to examine results of more alternative scenarios (or development) of the simulation. The original/main instance (one particular replication) of a simulation is cloned and individual clones have different parametrisation set for them. Conducting such nested simulation leads to simulating several alternative scenarios. The output of conducted nested simulations is a broader set of information about the given issue.

One possible use of nested simulations presents *decision support in a simulation*, the nested simulations run for a limited time and after they have finished, their results and outputs are assessed and the original simulation is again merged into a single instance (main simulation) and it can continue in a selected manner.

Another use of nested simulations can be realized as *a multi-trajectory simulation* - the simulation experiment is divided into nested simulations in individual points of decision and the simulation is gradually branching. Various scenarious are thus investigated and according to article (Gilmer and Sullivan 1999), such approach can be more efficient than using a large number of replications of a single simulation.

The following text deals with the first way of usage - as decision support in a simulation.

The use of nested simulations is also done by other authors. Use can be found, for example, in Bonté, Duboz, Quesnel and Muller (2009), Kindler (2010), Gordy and Juneja (2010). Individual publications are briefly presented in our previous article (Diviš and Kavička 2016).

2.1. The technique of a nested simulation

A nested simulation allows to use already existing simulation engine of a given software and a simulation experiment for own search of a solution of an issue. However, before the nested simulations can be conducted, it is necessary to list all steps solving a particular conflict (Diviš and Kavička 2016):

1) A conflict situation (an issue) requiring decision support is identified in the system.
2) Current instance of the main simulation (S^{main}) is interrupted in time t.
3) For the needs of nested simulation, it is necessary to set their parameters:
    a) the criterion of optimality (*CrOpt*),
    b) the duration of an outlook into the future for the nested trials (or rather the stopping condition, *StopCond*),

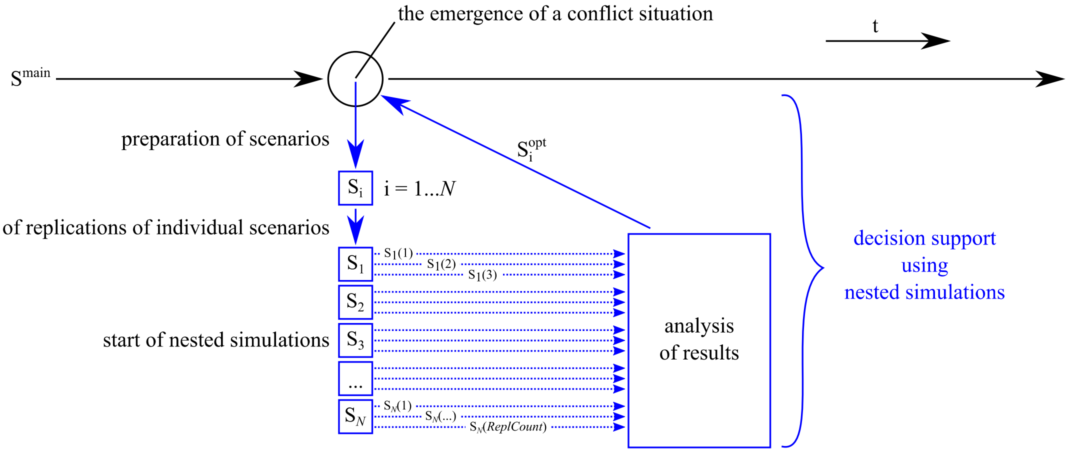


Figure 1: Illustration of occurrence and proceeding with solution of one conflict situation in one simulation replication

    c) the number of replications for all individual scenarios of the nested simulations (*ReplCount*),
    d) (maximal) number of alternative scenarios (*ScnCount*),
    e) generator of alternative scenarios (*ScnGen*),
    f) recursion limit of nested simulations (*RecLimit*).
4) N alternative scenarios for minor simulations are established.
5) The main simulation S^{main} is cloned and *ReplCount* of replications is created for each *i-th* scenario.
6) Individual replications $S_i(j)$ are started (for $i = 1…N$, $j = 1…ReplCount$).
7) Waiting for finishing all replications $S_i(j)$ (for $i = 1…N$, $j = 1…ReplCount$) to finish.
8) Assessing the results of individual scenarios from the replications $S_i(j)$ (for $i = 1…N$, $j = 1…ReplCount$) and then selecting the scenario with the best results according to *CrOpt.*
9) The main simulation S^{main} then continues with the selected scenario from the instant *t* of simulation time.

From the above described procedure (Figure 1) it is apparent that before the realization of nested simulations itself, it is necessary to solve several basic questions about how the nested simulations should even be parameterized and set up. A more detailed description of parameters of nested simulations can be found in article (Diviš and Kavička 2016).
A separate issue could be presented in how the alternative scenarios for solving conflict situations should be chosen, but such issue depends on particular application of a method and on the type of the conflict situation, and it can in the end present a non-trivial task.

2.2. Computational demands

Technical possibilities of nested simulation realization posed a separate and rather large issue for nested simulations. It could include a demanding computational task, complexity of which is given by the number of scenarios and replications and the length of simulated period of individual replications that need to be conducted.
Computational demand is influenced by several factors:

- the number of alternative scenarios,
- the number of replications of each scenario,
- the lenght of simulated period within one replication,
- the number of conflict states that occur in the original (main) simulation and that require a decision based on perspectives of the nested simulations,
- the number of replications of the original (main) simulation,

Apart from the abovementioned factors, it is also necessary to take into account the possibility of conflict situations occurring inside the nested simulations (example of recursive nested simulation is shown in figure 2). That could be cause by recursive launch of other nested simulations and thus cause an exponential growth in computation difficulty. One option how to avoid such an issue is to terminate the nested simulation exactly when a conflict situation occurs inside the nested simulation, or setting a maximal depth of recursion for nested simulation is a general solution.

3. DECISION SUPPORT WITHIN THE SCOPE OF SIMULATING RAIL TRANSPORT

In the general area of decision support, it is possible to use a variety of methods using even highly complex mathematical apparatus. From simple methods using priority lists to methods using artificial neural networks.
The role of decision support and its possible use in the environment of simulating rail transport is described in the next chapter.
To realize decision support in the environment of rail transport, it is basically possible to use all commonly used techniques including, for example:

- the method of priority planning,

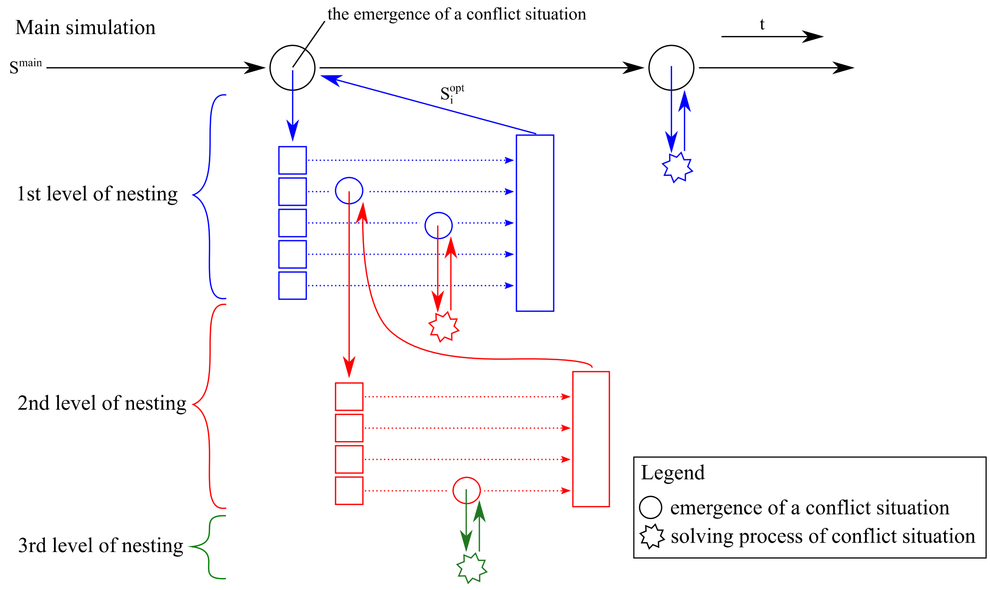


Figure 2: Illustration of a repeated occurrence of conflict situations during the main simulation and within nested simulations

- expert systems,
- methods of operational research,
- heuristic methods,
- methods of soft computing,
- the method of nested simulations,
- and others.

3.1. Methods used in simulators for rail transport

In the scope of microscopic simulators for rail transport, there are several different methods of decision support used.

Priority planning (or priority lists) is a rather common method of decision support. The principle of such method is to define a list of alternatives (e.g. train routes) that are ordered based on their priority. The decision is then done by choosing an option with the highest priority. Provided that such possibility is not permissible, the next item on the list is selected until all possibilities are exhausted. This method has been implemented in simulators OpenTrack and Villon.

More complex and more sophisticated system using the method of multi-criterion decision is newly implemented in the simulator Villon (Bažant and Kavička 2009).

3.2. Possible conflict situations, searching for alternative solutions.

Conflicts in stochastic simulations occur mainly as the result of individual entities (agents) competing with each other for limited resources. In the scope of the examined domain of rail transport, the identities are understood as trainsets competing for individual parts of infrastructure. Apart from the fact that there can be only one trainset present on one track, or rather a single train can be present on a single segment of switches, it is necessary to keep to defined rules of rail transport which are related to used safety device. Such rules also include safety time intervals with which it is not possible to allow another train to enter a track section in order to maintain safety and flow of traffic.

Specific examples of coflicts that can occur include, for example:

- Arrival of a train, requesting an already occupied station track, into the station.
- Gradual departure of more trains from the station (in a short time interval) while requiring the same line or part of the deviated tracks.
- Unscheduled train arrival into a stations with all platform lines occupied.

Searching for a solution depends on the aplication domain of the simulator. Within the scope of railway operation, it is possible to wait for clearing an occupied elements of the infrastructure. However, such solution usually is not suitable, it causes longer delays for the conflict train and it can cause other conflicts by occupying a line sector by the conflict train. An alternative solution is to move the train to another line or to another platform. It is impossible to easily predict whether such change would somehow affect the flow of traffic. In this case, nested simulations can analyze the impact of the given change on future traffic.

All solution possible to apply still need to keep to particular activities of safety device and all are limited by available infrastructure.

4. SIMULATION TOOL MESORAIL

The simulation tool *MesoRail* (Diviš and Kavička 2015) presents a mesocopic simulator of rail transport in development, which focuses on examining railway station capacity. The simulator specializes in support for

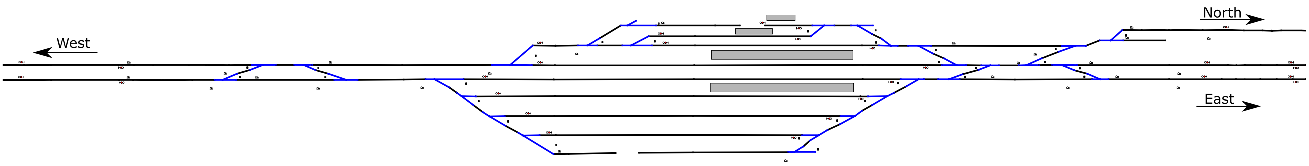


Figure 3: Infrastructure of center station in case study scenario

fast prototyping of rail infrastructure and fast setting of operational conditions specifications (including types and composition of trainsets, timetables, parameterization of random inputs, etc.). The aim of the simulator is to allow conducting simulation studies in shorter periods of time than it is possible with microscopic simulation tools. For these reasons, there are some abstractions implemented in *MesoRail* which sets it to mesoscopic level of abstraction. The main issue is finding a suitable compromise and level of abstraction for individual parts in such a way, that the required accuracy of result is achieved. To establish suitable abstractions, consultations with railway experts were conducted.

In the simulator the following abstractions were used - (i) schematic depiction of infrastructure, (ii) simplification of calculating train ride dynamics, (iii) neglect of mobile resources of operation (employees), (iv) neglect of some operational technological procedures. *MesoRail* can be applied for, for instance, determining the throughput of railway junctions, therefore it is necessary to meet the following requirements - (i) fast prototyping of infrastructure, (ii) real running features of trains, (iii) respecting the functionality of safety devices, (iv) applying station and track intervals, (v) simulating deterministic and stochastic train flows, (vi) animation outputs during the simulation, (vii) post-simulation statistics and time protocols in graphic form. The listed items is just a brief list of requirements expected form the simulator. Based on those requirements, the level of abstraction for individual components must be defined.

4.1. Architecture of simulation engine

Simulator *MesoRail* builds on hybrid architecture, which combines a discrete simulation engine with agent based simulation. The foundation is then a very simple engine using methods of event planning, which is, however, used in agent-oriented appoach.

4.2. Implementation of nested simulations

The method of nested simulations can make the impression that it is an easy and easily implemented task. However, its technical realization is rather challenging. For conducting of nested simulations, it is necessary to stop the original simulation, clone it, customize the clones' parameterization, and run the nested simulations for limited time.

Cloning a simulation means to create a complete copy of the current state of the simulation. For structured simulations deconstructed into agents (with inner state) that communicate among one another, it is necessary to create copies of agents, fill their state, and renew references among new instances of agents. If individual objects are realized as immutable, then creating a copy is easier. In *MesoRail* simulator, individual objects are - agents with inner state, and such state is changeable simulation, it is necessary to create a copy of a complex object graph.

After creating a copy, it is necessary to make available relevant agents and entities and to customize their parameterization for new simulation run. Launching the simulation as such is then a trivial task.

4.3. Technical realization

Because of above mentioned technical issues related to realizing nested simulations, general decision support for recording the state of object graph has been built in *MesoRail*. This system basically allows to arbitrarily go back in time of a finished simulation. Another function of this subsystem is creating a copy of an object graph and thus cloning the entire simulation.

Actual administration of nested simulation then efficiently uses other parts of the simulator for easy realization of a copy, parameter changing, and launching nested simulations.

4.4. Situations solved by nested simulations

Within the scope of realizing decision support, nested simulations are called in cases in which a train cannot allocate other line segments into its train path and stopping the train would be imminent. Such cases include, for instance, and occupied line at a station.

Individual trains have their train paths or groups of train paths defined, to which they keep druing the simulation. Nested simulations then use these alternative paths in group of train paths for finding a possible solution of a situation.

5. CASE STUDY

The following chapter describes realization of a case study using nested simulations as decision support in the scope of simulating a railway station. A simulation model contains one main prototype station (shown on figure 3), from which lead two double-lined (to the stations *West* and *East*) and one one-lined (to the station *North*) tracks. Those are finished by a simplified model of a railway station.

Total length of track from *West* to *East* is about 20 kilometers. The track contains also significant slope and arcs, track itself isn't completely artificial, but it is inspired by a several tracks in Czech Republic.

Table 1: Parameterisation of trains in the simulation model

Train type	Locomotive / vagons	Course	Interval between trains [h:mm:ss]	Total train count	Delay prob.	Delay mean time (exponential distribution) [s]
Express	1 / 7	West → Central → East	30:00	5	50%	420
Express	1 / 7	East → Central → West	30:00	5	50%	420
Passenger	2 / 4	West → Central → East	10:00	12	33%	270
Passenger	2 / 4	East → Central → West	10:00	12	33%	270
Passenger	1 / 2	West → Central → North	30:00	4	33%	270
Passenger	1 / 2	North → Central → West	30:00	4	33%	270
Cargo	1 / 22	West → East	1:00:00	2	50%	1800
Cargo	1 / 22	East → West	1:00:00	2	50%	1800

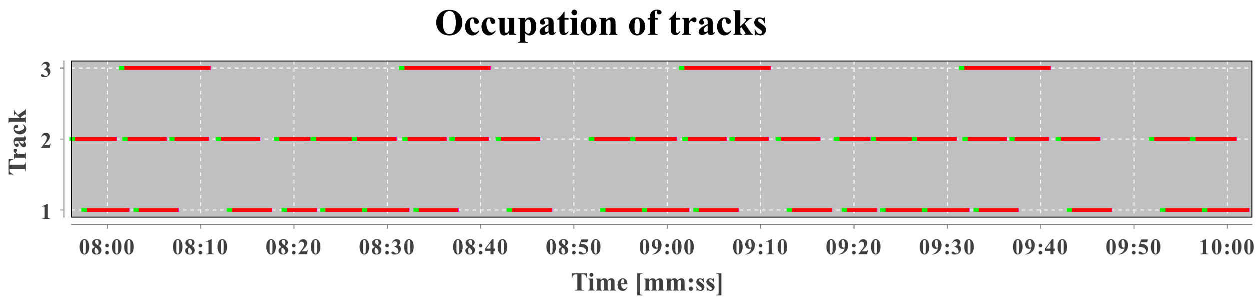


Figure 4: Occupation of station tracks during deterministic simulation

The traffic in station is mixed and it contains both passenger and cargo trains. It includes few express trains, dense traffic of passenger trains and several passing freight trains. A more detailed description of trains composition and parameterization is given in the table 1. The figure 4 shows the occupation of station tracks that are normally occupied by individual train sets under deterministic mode of operation.
Same simulation model was built also in the Villon for comparison of simulations results. The Villon represents microscopic simulation tool specializing in the construction of railway simulation models, but also supports road traffic and container logistics systems. The simulator works at a microscopic level of detail, and building a comparable model is more time consuming. In order to compare the results between Villon and MesoRail, it was first necessary to validate the results.

5.1. Validation of simulation models between MesoRail and Villon

After realization of the simulation models in both tools, deterministic simulations were performed without the use of random delays. During the simulations we recorded the moments of inputs and outputs of individual trains to or out of simulation. Based on these data, train travel times were calculated. The minimum travel time was 741 s, maximum 930 s, average 780 s. The results of the comparison of travel times are given in the table 2.

Table 2: Comparison of train travel times in MesoRail and Villon

	Relative difference [%]	Absolute difference $\lvert T_{MR} - T_V \rvert$ [hh:mm:ss]
Minimum	0,1285 %	00:00:01
Maximum	3,3735 %	00:00:31
Average	1,6200 %	00:00:13

The differences in the travel times of trains in both simulators is due to their different approach to train driving dynamics and minor deviations between both simulators and simulation models. The average size of difference is 1.62% and the maximum measured value is 3.37%. Since these values did not exceed the originally assumed limit of 5%, both models were considered to be identical and so it was proceeded to stochastic testing.

5.2. Stochastic simulations

When making stochastic simulations, every train entering the model may be delayed. First, using a random value generated with a uniform distribution, it is determined whether the train will be delayed or not. The amount of delay is then given by random number

Table 3: Evaluation of simulation results - statistic of change of train delays after simulation run

Type	Absolute delays [s]			Relative delays [%]		
	Villon	MesoRail 5 min	MesoRail 15 min	Villon	MesoRail 5min	MesoRail 15min
Express	40,61	14,79	14,27	5,16%	1,90%	**1,83%**
Passenger	40,56	24,42	26,93	5,25%	**3,16%**	3,49%
Cargo	35,50	6,96	11,59	3,95%	**0,76%**	1,27%
Total	40,13	20,81	22,84	5,11%	**2,68%**	2,94%

Table 4: Statistics of conflicts during one replication of the main simulation

Outlook duration	5 min			15 min		
	Min	Max	Average	Min	Max	Average
Conflicts in main simulation	3	15	7,88	3	18	8,04
Total number of conflicts	3	89	17,19	3	432	67,79
Total simulations ran	4	172	31,94	4	843	119,78

generator with exponential distribution (according to the type of train type). No further random effects (device faults, station delays, ...) are being used during the simulation.
Each train has a defined starting train path and three alternative train paths that use other station tracks. The strategy used to select a replacement train path is given by the simulator.
Villon uses the priority planning technique. If the station track is occupied, then based on the order of paths in priority list, first available alternative path is selected. Therefore, only the order of the defined routes is important for the selection of the alternate train path and whether or not they are available at the moment, there is no further analysis of the conflict situation.
MesoRail uses nested simulation techniques. In the event of a conflict situation, individual alternatives are simulated for a limited time period. The number of replications of nested simulations, the length of their run is described in the following section. To test the effect of the parameters, two sets of simulations were performed with different outlook values (the duration of the nested simulation).
The duration of one simulation replication is 2.5 hours, all the trains are scheduled to run during this period. In order to test different variants and to obtain data for processing statistics, 100 replications were performed in Villon and 100 replications in MesoRail.

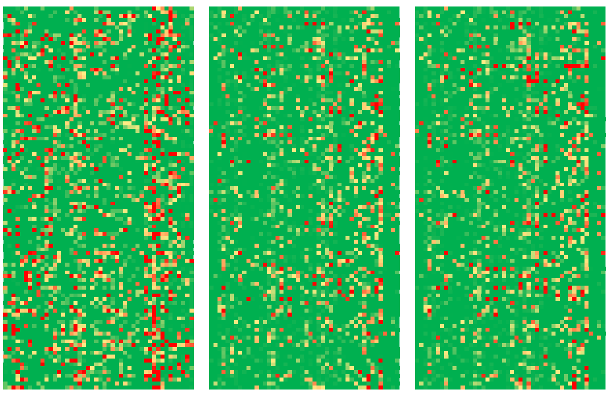
5.3. Parameterisation of decision support

Villon has defined paths according to the expected priorities appropriate for the type of train and the direction in which it is going.
Parameterization of nested simulations in MesoRail is as follows:

- Optimality criterion - the sum of weighted train delays, the weight represents the train's priority according to its type (express trains – 1.8, passenger – 1.0, cargo – 0.2);
- Length of outlook - 5/15 minutes of simulation time from the moment of conflict;
- Number of replications of nested simulations - 1 replication;
- Number of alternative scenarios - not limited;
- Alternative scenario generator - according to available alternative train paths;
- Recursion limit of nested simulations - up to 3 levels.

5.4. Simulation results

At the end of the simulations, the results were analyzed. As a basic indicator of the quality of the operation was chosen the change in the delay of the train (calculated as delay at the moment of leaving simulation minus train's input delay). Due to the fact that the train timing in both simulation models is not exactly the same, the results are evaluated not only in the form of the absolute value of the delay and also the relative delay given by the ratio of the absolute value of the delay to the travel time of the train. The aggregate results for each train type



(a) Villon (b) MR 5 min (c) MR 15 min

Figure 5: Heatmaps of relative delay changes (vertical axis – individual replication, horizontal axis – individual trains, green – no delay, yellow – 10% delay change, red – 30% delay change)

ISBN 978-88-97999-85-0; Affenzeller, Bruzzone, Jiménez, Longo and Piera Eds.

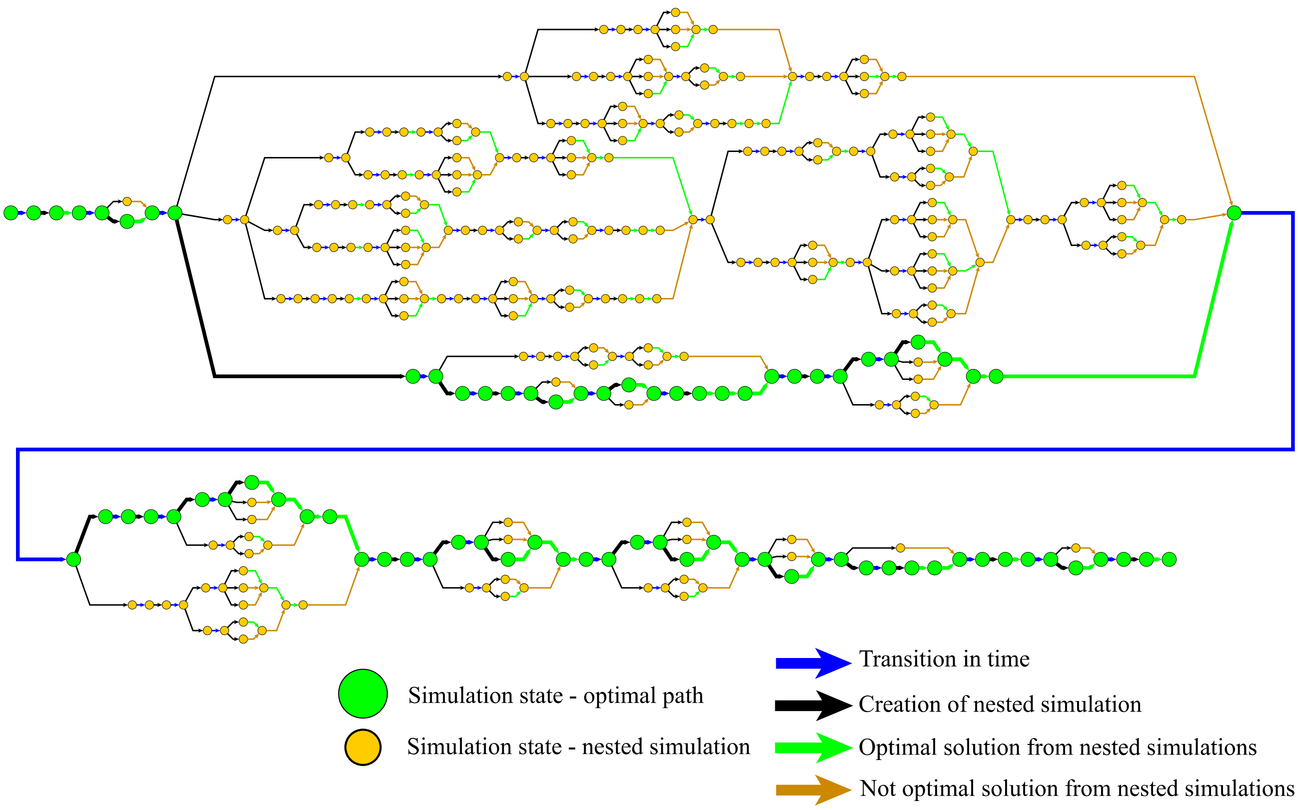


Figure 6: Simulation state diagram – main replication nr. 13, outlook length 5 min (11 conflicts in main simulation, 89 conflicts total, 172 simulations ran)

and the overall results of the results are presented in the table 3. Delays of individual trains by replication are shown in the image 5.

The results show that the technique of nested simulations achieves better results than the simple method of priority planning. Due to the differences between the two simulation models in Villon and MesoRail, it is not possible at this time to say that the improvement was exactly 50% in terms of delays of simulation trains. However, results from nested simulations suggest that use of individual tracks is more appropriate and leads to minimization of subsequent conflicts.

5.4.1. Analyze of nested simulations

Comparison of the two parameterisations of nested simulations with 5-minute and 15-minute outlooks did not entirely go as expected. A longer perspective gave slightly worse average results than a shorter variant. This is probably due to the higher priority of the express trains compared to other trains and the greater emphasis of the algorithm on the effort to optimize their travel times.

The statistics of the number of conflicts that occurred in the main simulation and in the nested simulations and the total amount of simulations performed within one simulation replication can be seen in the table 4. On the i7 quad-core test set, on average, one major replication with 5 minutes outlook needed 11 minutes (of real time) to complete the calculation, for a 15 minute outlook, the average calculation time was 35 minutes.

Figure 6 shows a state diagram of replication nr. 13 (5 minute outlook). Here are all the moments when the nested simulations were executed and the resulting optimal path is shown. This is an example of one of the more complex replications, with many recursive calculations. Simulation of this replication took about 20 minutes of real time.

Figure 7 shows the MesoRail simulator during calculation of replication nr. 79 (15-minute outlook). During the simulation, it is possible to display the status of each running simulation and the state diagram is automatically generated.

6. CONCLUSION

The method of nested simulations suitable for use as a tool for multi-trajectory simulation or as decision support in a simulator, with focus placed on the latter, has been presented.

The method has been implemented within the simulator MesoRail. It presents a mesoscopic simulator of rail transport with focus to test throughput of train nodes.

Nested simulations are tested on a complex case study of a railway station. The simulation results show that the nested simulation technique allows a better evaluation of the alternate station track rather than a simple priority planning method. Due to differences in the microscopic and mesoscopic approach, it can not be

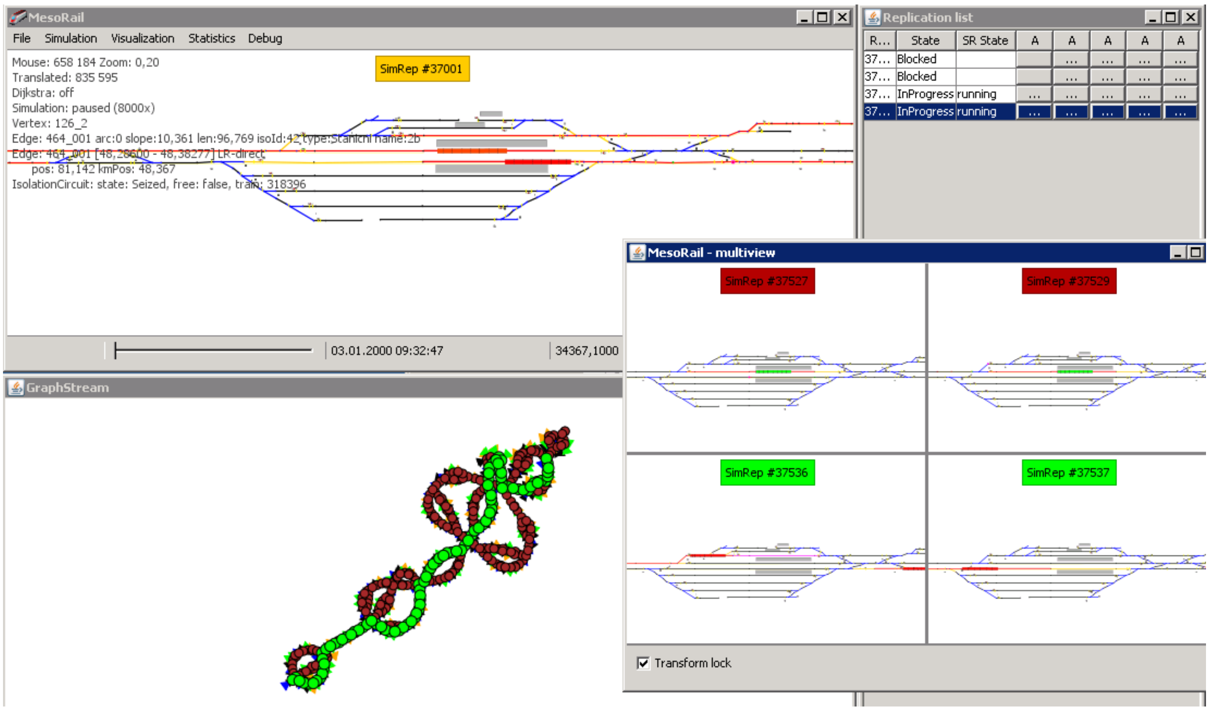


Figure 7: MesoRail tool during simulation (replication nr. 79, 15 minute outlook)

said that the improvement has reached a specific numerical value.

Techniques of nested simulations can be used as a general method of decision support within any application domain. However its implementation use is not quite trivial, and the computational demand of the method is also not negligible.

REFERENCES

Bažant M., Kavička A., 2009. Artificial neural network as a support of platform track assignment within simulation models reflecting passenger railway stations. Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit Vol. 223(5) : 505-515.

Bonté B., Duboz R., Quesnel G., Muller J. P., 2009. Recursive simulation and experimental frame for multiscale simulation. In: Proceedings of the 2009 Summer Computer Simulation Conference, 164-172. July 13-16, Istanbul, Turkey.

Diviš R., Kavička A, 2015. Design and development of a mesoscopic simulator specialized in investigating capacities of railway nodes. Proceedings of the European Modeling and Simulation Symposium, 52-57. September 21-23, Bergeggi, Italy.

Diviš R., Kavička A, 2016. The method of nested simulations supporting decision-making process within a mesoscopic railway simulator. Proceedings of the European Modeling and Simulation Symposium, 100-106. September 26-28, Larnaca, Cyprus.

Gilmer J. B., Sullivan F. J., 1999. Multitrajectory simulation performance for varying scenario sizes [combat simulation]. In: WSC'99. 1999 Winter Simulation Conference Proceedings. 'Simulation - A Bridge to the Future' (Cat. No.99CH37038), 1137-1146. December 5-8, Phoenix, AZ, USA.

Gordy M. B., Juneja S., 2010. Nested Simulation in Protfolio Risk Measurement. Management Science 56:1833-1848.

Kindler E., 2010. Nested Models Implemented in Nested Theories. In: Proceedings of the 12th WSEAS International Conference on Automatic Control, Modelling & Simulation, 150-159. May 29-31, Catania, Sicily, Italy.