32nd European Modeling & Simulation Symposium
17th International Multidisciplinary Modeling & Simulation Multiconference

ISSN 2724-0029 ISBN 978-88-85741-44-7 © 2020 The Authors.
DOI: 10.46354/i3m.2020.emss.005

Train overtaking at railway stations within simulation models of railway lines

Tomáš Vyčítal[1,*] and Michael Bažant[1]

[1]University of Pardubice, Department of Software Technologies, Studentská 95, Pardubice 532 10, Czech Republic

[*]Corresponding author. Email address: tomas.vycital@student.upce.cz

Abstract

Nowadays there are available simulation tools that can be used to simulate railway traffic but not that much attention has been paid to the quality and intelligence of some decision support systems used to resolve possible conflicts during simulation experiments. Objective of this work is to investigate more complex decision support system in the area of train overtaking within railway stations and to assess the level of benefits that this approach brings. The presented algorithm is valid in general conditions, but it is also validated in connection with simulation tool OpenTrack which is one of the leading simulation tools in the area of railway traffic simulations focused especially on railway lines. Possible solutions to this problem that are available in OpenTrack are considerably less sophisticated and their configuration is often not very intuitive. This topic is also discussed in greater detail in the paper. To show the advantages of this advanced decision-making approach, a case study that compares common and advanced decision-making approaches is presented.

Keywords: simulation of railway traffic; decision-making support; train overtaking

1. Introduction

Overtaking is not very intuitive or easy to configure in simulation models of railway traffic. Often it's really difficult or even impossible to achieve such overtaking that would closely resemble real world situations without programming custom algorithms to make the decisions. This commonly results in less than ideal behavior of trains in the simulation. Common pitfalls include slow commuter or freight trains blocking express trains, trains waiting on the mainline to be overtaken, overtaking over speed-limited sidings or trains waiting at sidings for overtaking that never happens. If not configured carefully it can even create deadlocks (two or more trains waiting for each other to depart). This generally can be avoided with well thought configuration, although some compromises may be necessary. Alternatively the deadlocked or otherwise misbehaving runs can be simply thrown away and ignored.

The work described in this paper tries to put dispatcher's way of thinking into formal algorithm, implement it in code and use it to resolve conflicts in a simulation. The main focus is to replicate the mental process of real world human dispatchers in a running simulation. The other important requirements are the ease of use, intuitive configuration and easy extendability of the algorithm.

To show and verify the merits of this approach, an application was developed and used with the simulation tool OpenTrack on a part of the mainline through Pardubice (a town in the Czech Republic, over 50 km of tracks). The results were then compared to stock OpenTrack simulation runs.

2. State of the Art

Broadly speaking train rescheduling is a well researched topic with many interesting ideas being explored and improved upon. However as of our best knowledge, this specific case haven't been properly explored so far. The work that we found focuses mostly on finding the best rescheduling results possible, creating the ideal timetable, optimizing algorithm performance on computers etc. This is of course not wrong and it is both interesting and useful in many real world scenarios, but it may not necessarily be desirable for example in a simulation trying to closely replicate infrastructure and traffic governed by human dispatchers. In this case even better means wrong as it doesn't correspond well enough to the decisions made by actual dispatchers in similar situations.

Various methods have been explored on closely related topics. For example max-plus linear model algebra is described in A permutation-based algorithm to optimally reschedule trains in a railway traffic network van den Boom, T.J.J. and Weiss, N. and Leune, W. and Goverde R.M.P. and De Schutter, B. (2011). The solution however assumes many simplified conditions, e.g. stations have enough capacity for all considered trains, periodic timetable, fixed headway times etc. Due to these assumptions this isn't general purpose solution, at least not without further work.

In Hernando et al. (2009) part of the presented project for Spanish railway operator (RENFE) is focused on taking decisions about lengthening and passing loops. The presented solution is focused on similar problem but with aim to find passing loops in the model that could be interesting to lengthening. However there are only brief information about conditions that must be met but there is no information how the algorithm consider headway calculations, train categories, time intervals between trains etc. The algorithm is implemented in proprietary simulation tool "Graph Viewer", so no other information could be found.

In the paper Train Timetable Optimizing and Rescheduling Based on Improved Particle Swarm Algorithm Meng, X. and Jia, L. and Qin Y. (2010) good results have been obtained using an algorithm based on particle swarm. The authors especially focused on maximizing timetable stability through rescheduling of trains. They achieved good results with this method, though they didn't attempt to mimic human dispatchers, they tried to achieve even better results than humans generally do.

The very same intention of superhuman dispatching has been explored in Deep Reinforcement Learning Approach for Train Rescheduling Utilizing Graph Theory Obara et al. (2018) utilizing graph theory and neural networks. Even though they used simplifications in their experiments and had some performance problems that at the moment rule out real-time use in complex railway systems, the method showed good results with small-scale model and looks promising as an improvement over human dispatching. Again this isn't exactly in line with the aim of our research.

Another area explored by many researches is the otimization of computational performance. For example depth-first search with solution space represented using binary trees has been proposed, implemented and tested in A parallel algorithm for train rescheduling Josyula et al. (2018). Considerable improvements has been reported especially in performance and reaction times. The use of GPUs to offload some computation and gain performance benefits has also been explored in Exploring the Potential of GPU Computing in Train Rescheduling Josyula et al. (2019) although to limited success when compared to purely CPU based computations. As suggested by the paper moving the entire computation on the GPU instead of just offloading parts of it looks promising, thought this is still an open question. The methods used in these papers, though reportedly very efficient on computers and prividing good dispatching, don't closely resemble the mental process of a real world dispatcher, who is unlikely to use given concepts or even be able to comprehend them.

2.1. Stock OpenTrack

Special attention is paid to stock OpenTrack (i.e. the basic simulation tool without the use of any extensions through it's API). The reason is that OpenTrack is popular tool for simulation of traffic on railway lines and many scientists and simulation experts use it without any custom algorithms at all.

In stock OpenTrack overtaking can be tweaked by the combination of dispatching policy, priority, look ahead distance and timetable tweaks (e.g. train stopping on a siding for some time hoping that another train will pass in the meantime) Huerlimann, D. and Nash, A. B. (2017). This is really cumbersome to set up. If the look ahead distance is set too long, slow trains will be blocked too early and wait for overtaking, even though they could continue, reach the next station and be overtaken there without blocking any faster trains in the meantime. If too short, faster trains will often get stuck behind slower trains and get unnecessarily delayed. It's made even worse by the fact that look ahead distance is the same throughout the model as it can only be configured per train category. Therefore it may not even be possible to find such a value that would work well throughout the whole model. If there are long railway line segments without overtaking opportunities in some parts of the model and closely packed overtaking opportunities in other parts, the look ahead distance is inevitably going to be too short, too long or both at the same time. This leads to suboptimal and unrealistic overtaking in the simulation.

3. Train overtaking

Nowadays timetables are constructed using various software systems considering capacity of railway lines, railway stations etc. In case of unperturbed railway operations is everything clear. Unfortunately, more common is perturbed traffic where method of computer simulation is useful for:

- investigation of various scenarios
- investigation of consequences of delays
- robustness of timetables testing

To obtain valuable results from simulation model it is important to implement all substantially decision problems with aim to reach similar decisions as in reality.

When train delays occur, it is important for a dispatcher to consider headway calculations in order to select appropriate sequence of trains that run the same railway line in order to avoid slowing down faster train. Dispatchers have an aid in the form of a table where all train type pairs are listed with a value of necessary time spacing.

When dispatchers detect situations that two trains, that will use the same output track from a railway station, are arriving to a railway station and time spacing between them is less than needed headway, it is important to change sequence of these trains.

In principle, the same behavior is expected in simulation tools where it is possible to examine time interval between trains and as it changes during simulation execution. Unfortunately, it is not that easy to implement this kind of decision making because quite a lot of input data is needed, and implementation of all needed rules is not straightforward within a simulation tool.

In OpenTrack there is available solution to this problem by dispatching priority for categories of trains. For each train category user can define look ahead zone (in terms of distance or time) that is monitored during simulation and if a train with a lower priority is detected then this train is overtaken at a railway station Huerlimann, D. and Nash, A. B. (2017).

4. Algorithm

The basic idea behind the algorithm used is to consider two points in the model of infrastructure. The first point is where there is an opportunity for overtaking and the second point is where given trains either part or have another opportunity to overtake each other. The trains approaching the first point are then examined based on their expected times (including current delays) of arrival at the second point in the model. If they are expected to arrive at the second point in the same order, there's no problem and no overtaking will be performed. However, if that's not the case the slower trains which would block the faster ones (potentially there might be more faster trains blocked with slower train) will be instructed to stop at the first point and let the faster trains pass.

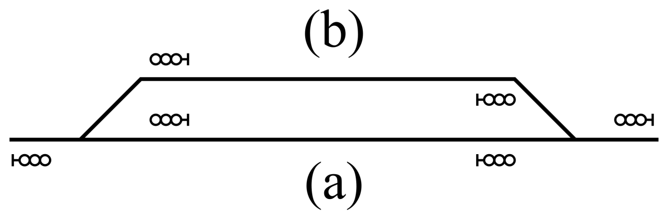

Figure 1. Overtaking opportunity
The track (a) where the overtaking train passes. The track (b) where the overtaken train waits to be overtaken.

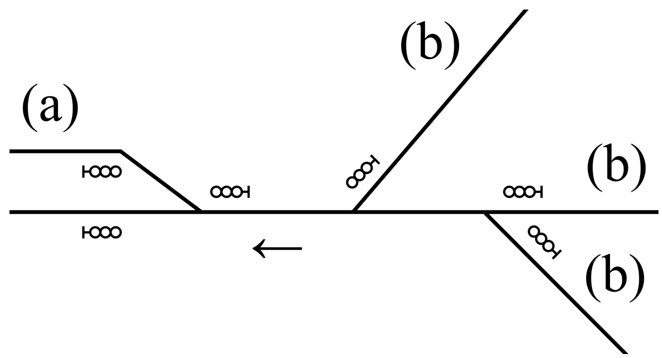

Figure 2. Overtaking area with multiple input tracks
The overtaking opportunity (a) this area terminates at. Different input tracks (b) all joining in this overtaking area and continuing to the overtaking opportunity (a).

4.1. Overtaking opportunity

This term refers to a track (a in Figure 1) in the model that has one or more sidings (b in Figure 1) and has been designated for overtaking. Slower trains will be then instructed to wait on the siding until the faster trains pass.

4.2. Overtaking area

This is an area where overtaking is considered and that terminates at a single overtaking opportunity where the overtaking can actually take place. The incoming trains can come from different input tracks that all join prior to the overtaking opportunity. The input tracks may also contain other overtaking opportunities. This is handy especially in situation where there are multiple closely packed overtaking opportunities prior to a longer stretch of tracks without any opportunity for the trains to overtake one another, especially if there are railway stops where regional trains may stop and

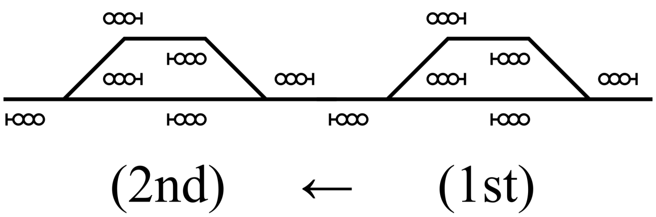

Figure 3. Two consecutive overtaking opportunities

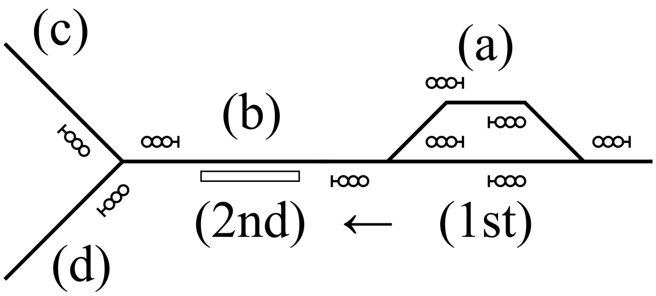

Figure 4. Overtaking opportunity with multiple outflow tracks
The overtaking opportunity (a) that will be used as the first point and the station (b) as the second by the algorithm. The trains will pass the station (b) and then may part ways using the outflow tracks (c) and (d).

block intercity traffic. The trains may also part again after leaving the overtaking opportunity, but they have to meet there for the overtaking to be possible. Only the trains that find themselves in the overtaking area are considered for overtaking at its overtaking opportunity.

4.3. Decisions

The decisions, as has been stated in the introduction to this chapter, are based on the order in which the trains that are going to pass through one overtaking opportunity are expected to arrive at some second point in the model. The second point is determined individually for each pair of trains that are being considered as candidates for overtaking. It can either be the next overtaking opportunity that they're both going to pass or some other point after which they're going to part ways.

The following requirements have to be fulfilled to plan and perform the overtaking:

- At the overtaking opportunity, there has to be space for the train to stop and wait to be overtaken (suitable empty track).
- At the second point, the scheduled arrival time of both trains has to be known.
- Current positions and delays of both trains have to be known.

If at least one of these requirements is not fulfilled the decision is to not overtake.

The expected order of arrival at the second point is based mainly on the arrival time from the timetable with some further adjustments based on the current state in the model. Current delay of each train is used to offset the planned arrival to get more accurate expected arrival time. In case that there is some reserve in the timetable between the position of the train and the second overtaking opportunity it is assumed that it will be used to reduce the delay (e.g. if a train is running 5 minutes late and is supposed to stop in a station for 10 minutes but with minimum dwell time of only 2 minutes, it's assumed it will stop for 5 minutes and depart on time). Considering all this information the expected times of arrival at the second point can be computed and evaluated.

The decision-making algorithm then considers whether the first train is expected to arrive first or whether the second train would arrive first, if the first train didn't block its way. If the first train would be first or they are expected to arrive at the same time, then overtaking doesn't make any sense and won't be performed. If the second train is expected to arrive first though, the time difference between their arrivals will be evaluated. The reason for further evaluation is that if the difference is very small (i.e. the trains would theoretically arrive at about the same time, if they could occupy the same space at the same time) the act of overtaking may cause a lot of additional problems, but not come with many or even any benefits. For this reason, a threshold is introduced. If the difference doesn't exceed the threshold, the trains won't be instructed to overtake one another even though they are expected to arrive in the wrong order. Only if the second train is expected to be faster enough than the first train to exceed the threshold, the first train will be stopped at the overtaking opportunity and wait for the second train to pass and only when the second train passed, the first train will be released and depart from the overtaking opportunity. The trains that, according to the timetable, were not supposed to stop at given overtaking opportunity will be stopped there anyway in order to perform the overtaking.

The same logic is being used to perform overtaking for all overtaking opportunities in the model and for all relevant pairs of trains. Therefore, a train can overtake and be overtaken multiple times during its journey within the model or at a single overtaking opportunity. Also, if for example a train managed to get from the first overtaking opportunity to the second overtaking opportunity without blocking any other train but won't make it to the third it will be stopped at the second and wait for the faster train to pass.

The decision-making process is also illustrated using flowcharts in figures 7, 8 and 9

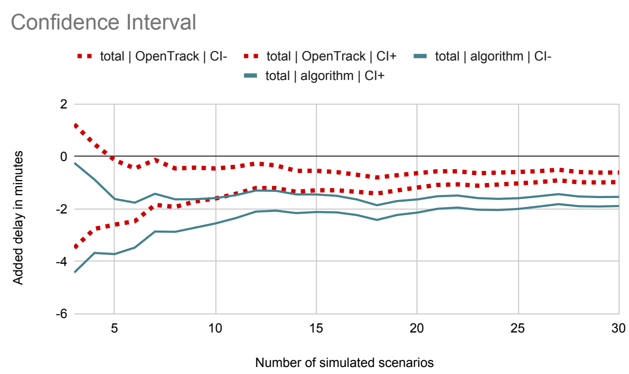

Figure 5. Begin to end delay difference

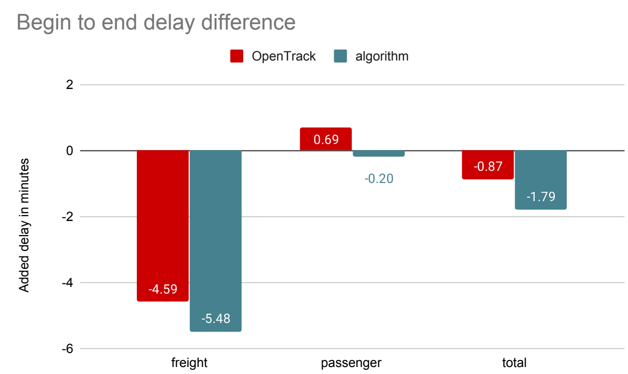

Figure 6. Delay difference per train category

5. Methods

Model of the mainline through Pardubice has been built in OpenTrack in order to test the algorithm with real world infrastructure and traffic. The model is centered around the town Pardubice in the Czech Republic with over 50 km of mainline plus a short branch line through Rosice joining with the mainline in Pardubice. Thanks to the branch line trains are leaving and entering the mainline at different locations and also crossing the mainline tracks to get from the southern tracks of Pardubice main station to Rosice, which is to the north. The model contain 2 h of traffic based on data obtained from SŽ (local railway infrastructure manager) whereas the timetable is organized as theoretical concept that is investigated within the simulation model. The delays were configured according to SŽDC SM124 guideline SŽDC SM124: Zjišťování kapacity dráhy (2019) (up to 2 h on entry).

The same configuration was used with and without the algorithm, each with 30 different delay scenarios (the same scenario was always used with both methods). Begin to end delay differences per train category were measured and compared. The general behavior (e.g. whether waiting trains block the mainline or use the sidings) was also observed, though it wasn't quantified.

Dispatching in the simulation was set to default (FIFO) with 0 m look ahead distance for all train categories and in all runs. Compared with other possible values, these didn't cause deadlocked runs (see ?? section for more details about such problems). As for the configurable threshold of the algorithm, 4 min threshold has been picked and used for comparison.

6. Results and discussion

As can be seen from Figure 5 there was significant decrease in begin to end delay difference in overall traffic (given 95 % confidence interval) with the algorithm when compared to stock OpenTrack. There were significant differences between passenger and freight traffic as shown on Figure 6. Freight trains on average decreased their delays more with the algorithm (about −5.5 min difference) than with stock OpenTrack methods (about −4.5 min difference). Passenger trains originally had higher delays on average when leaving the model than on entry (about +36 s difference). The algorithm managed to cut some of the delay (about −20 s difference).

Thanks to the fact that the algorithm bases it's decisions on the actual positions, delays etc. of trains, overtaken trains almost always stop and wait on sidings rather than on the mainline. In stock OpenTrack it's necessary to pick the right look ahead distance for this, which may not always be possible.

The algorithm is, given it's formulation, also very easy to adapt to specific situations. For example, all lines in the model are electrified. However if there are used only partially electrified lines, the "suitable empty track" requirement to plan overtaking from the algorithm can check whether given track has overhead lines for an electric train and if not simply evaluate to false and send the train elsewhere or abort the overtaking altogether if no suitable tracks are found.

7. Conclusions

The algorithm avoids faster trains trailing slower trains (e.g. fast intercity passenger train being stuck behind slow freight train) noticeably more than stock OpenTrack does.

The fact that the trains tend to stop on sidings for overtaking without blocking the mainline more than with stock OpenTrack is also closer to reality. A real world dispatcher would try to avoid having faster trains overtake using sidings due to much higher speeds generally allowed on the mainline when compared to sidings and also longer distance the train using siding usually has to cover. Overtaking over sidings is especially problematic if the overtaking train doesn't stop there, though it's not exactly ideal even if it's a station where the overtaking train normally stops.

The algorithm also automatically adapts to the infrastructure it's used on. This removes the problems associated with choosing the right (or usually least wrong) look ahead distance for train categories and other related configuration. It is therefore much easier to use, especially for people not well versed in given simulation tool.

References

Hernando, A., Roanes-Lozano, E., and García-Alvarez, A. (2009). An accelerated-time microscopic simulation of a dedicated freight double-track railway line. In *Mathematical and Computer Modelling 51 (2010)*, pages 1160–1169.

Huerlimann, D. and Nash, A. B. (2017). *OpenTrack Simulation of Railway Networks Version 1.9*. OpenTrack Railway Technology Ltd. and ETH Zurich Institute for Transport Planning and Systems.

Josyula, S. P., Törnquist Krasemann, J., and Lundberg, L. (2018). A parallel algorithm for train rescheduling. *Transportation Research Part C*, 95:545–569.

Josyula, S. P., Törnquist Krasemann, J., and Lundberg, L. (2019). Exploring the potential of gpu computing in train rescheduling. In *8th International Conference on Railway Operations Modelling and Analysis*.

Meng, X. and Jia, L. and Qin Y. (2010). Train timetable optimizing and rescheduling based on improved particle swarm algorithm. *Transportation Research Record: Journal of the Transportation Research Board*, 2179:71–79.

Obara, M., Kashiyama, T., and Sekimoto, Y. (2018). Deep reinforcement learning approach for train rescheduling utilizing graph theory. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 4525–4533. IEEE.

SŽDC SM124: Zjišťování kapacity dráhy (2019). *SŽDC SM124: Zjišťování kapacity dráhy*. SŽDC, Prague. Table 3, page 33.

van den Boom, T.J.J. and Weiss, N. and Leune, W. and Goverde R.M.P. and De Schutter, B. (2011). A permutation-based algorithm to optimally reschedule trains in a railway traffic network. In *Proceedings of the 18th World Congress*, pages 9537–9542.

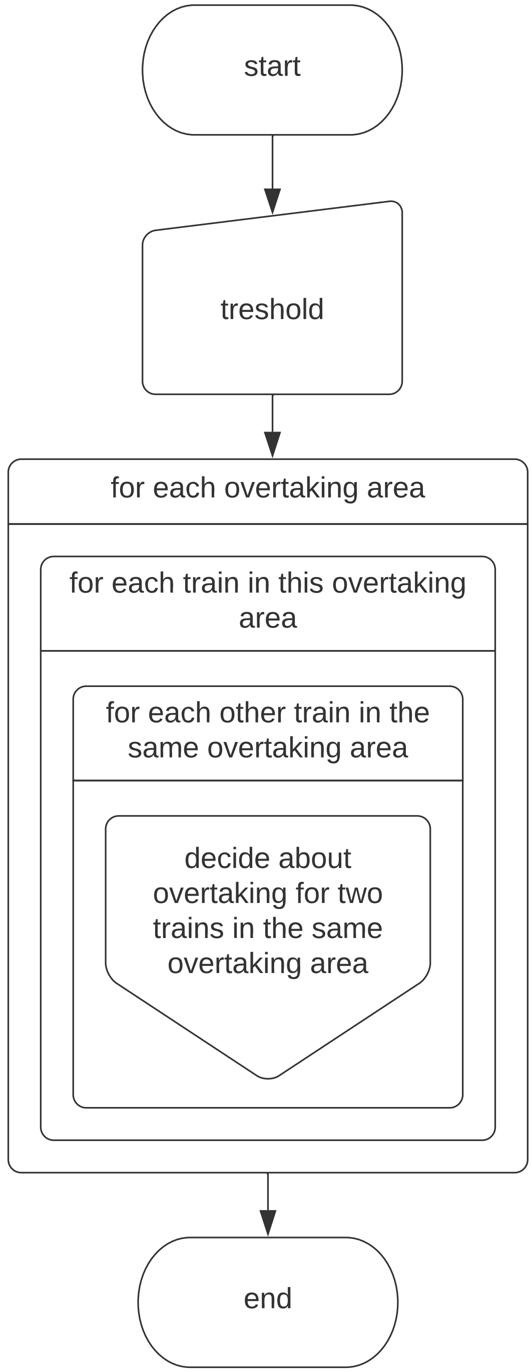

Figure 7. Flowchart: plan or cancel overtaking

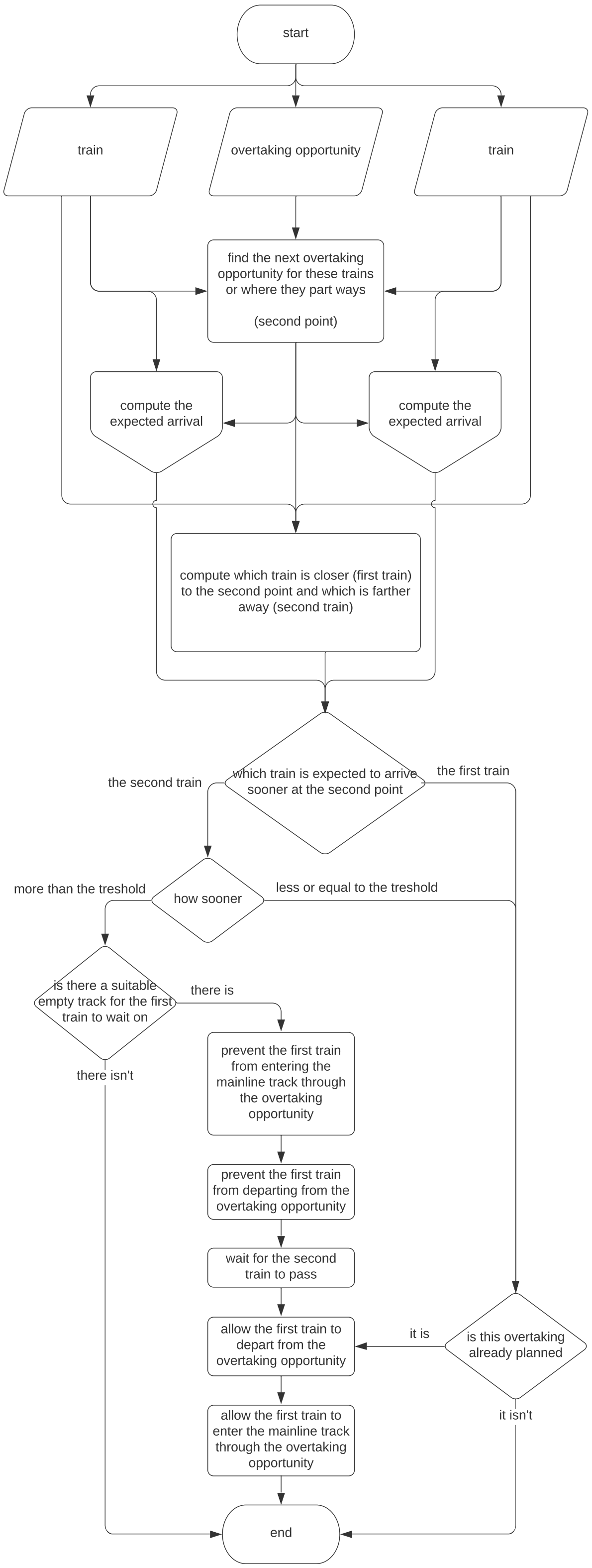

Figure 8. Flowchart: decide about overtaking for two trains in the same overtaking area

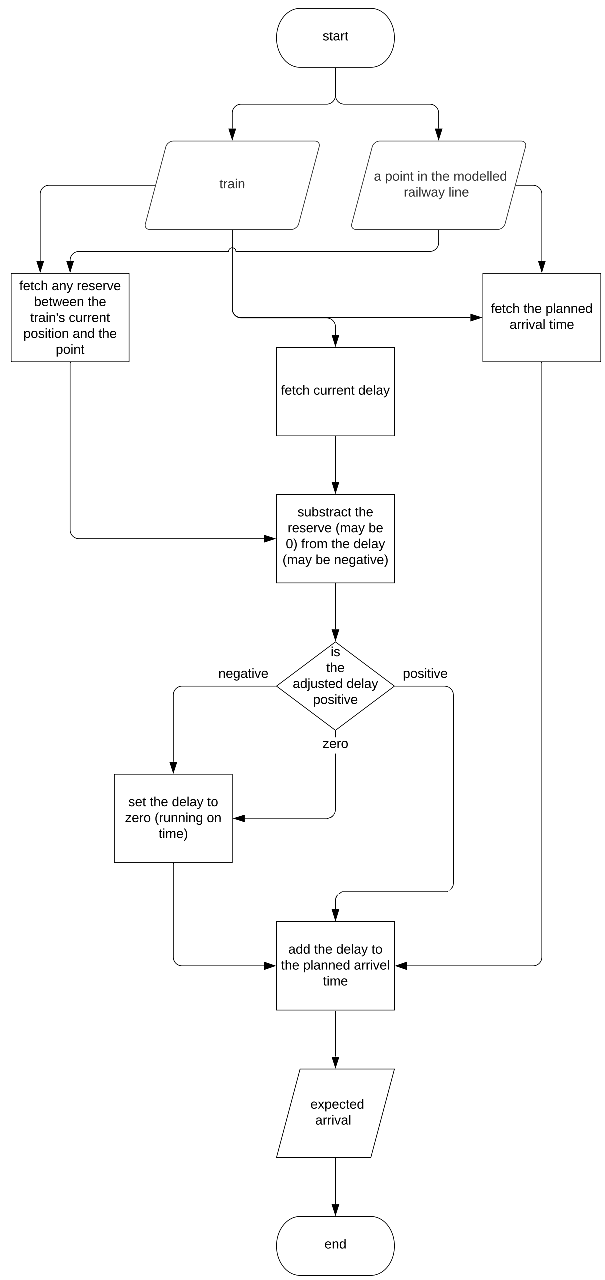

Figure 9. Flowchart: compute the expected arrival