TORS
A train unit shunting and servicing simulator

Van Der Linden, Jacobus G.M.; Mulderij, Jesse; Huisman, Bob; Den Ouden, Joris W.; Van Den Akker, Marjan; Hoogeveen, Han; De Weerdt, Mathijs M.

Publication date
2021

Document Version
Final published version

Published in
20th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2021

Citation (APA)
Van Der Linden, J. G. M., Mulderij, J., Huisman, B., Den Ouden, J. W., Van Den Akker, M., Hoogeveen, H., & De Weerdt, M. M. (2021). TORS: A train unit shunting and servicing simulator. In *20th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2021* (pp. 1773-1775). (Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS; Vol. 3). International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS).

TORS: A Train Unit Shunting and Servicing Simulator

Demonstration Track

Jacobus G.M. van der Linden
Delft University of Technology
J.G.M.vanderLinden@tudelft.nl

Jesse Mulderij
Delft University of Technology
J.Mulderij@tudelft.nl

Bob Huisman
NS, Utrecht
Bob.Huisman@ns.nl

Joris W. den Ouden
NS, Utrecht
Joris.denouden@ns.nl

Marjan van den Akker
Utrecht University
J.M.vandenAkker@uu.nl

Han Hoogeveen
Utrecht University
J.A.Hoogeveen@uu.nl

Mathijs M. de Weerdt
Delft University of Technology
M.M.deweerdt@tudelft.nl

ABSTRACT

When trains are finished with their transportation tasks during the day, they are moved to a shunting yard where they are routed, parked, cleaned, subject to regular maintenance checks and repaired during the night. The resulting Train Unit Shunting and Servicing problem motivates advanced research in planning and scheduling in general since it integrates several known individually hard problems while incorporating many real-life details. We developed an event-based simulator called TORS (Dutch acronym for Train Shunting and Servicing Simulator), that provides the user with a state and all feasible actions. After an action is picked, TORS calculates the result and the process repeats. This simulator facilitates research into a realistic application of multi-agent path finding.

KEYWORDS

Event-based simulation; Railway Operations; Train Unit Shunting Problem; Multi-Agent Path Finding; Train Unit Shunting and Servicing; TORS

ACM Reference Format:
Jacobus G.M. van der Linden, Jesse Mulderij, Bob Huisman, Joris W. den Ouden, Marjan van den Akker, Han Hoogeveen, and Mathijs M. de Weerdt. 2021. TORS: A Train Unit Shunting and Servicing Simulator: Demonstration Track. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 3 pages.

Demo information: Git: https://github.com/AlgTUDelft/cTORS, Link to video demo.

1 INTRODUCTION AND BACKGROUND

NS (Dutch Railways) is the main train passenger transporter of the Netherlands. A major asset of the NS is its fleet of trains. At network nodes or hubs, consisting of a major station and associated shunting yards, an important process is executed to deliver the physical trains for passenger transportation duties. After each rush hour there is a return flow of trains leaving the transportation process. In between the trains receive 1st-line service (internal and external cleaning, small repairs), their composition can be modified, and they are parked. The overall input and output flow of a yard is specified by the timetable. Environmental public transportation by rail is expected to remain growing for the next decades. However, most hubs are located in dense urban areas where space is very limited, while the pressure to reduce cost increases. The common challenge is to enable more transportation volume without extending the rail infrastructure significantly and to offer robust services to passengers. Among other measures, it requires the ability to plan and schedule node processes based on real-time information about the transportation process and fleet status.

The planning problem of shunting the trains and scheduling the service tasks is currently in large part solved by human planners. The practical problem consists of many elements and is often explained as the Train Unit Shunting and Servicing (TUSS) problem [10, 11] which consists of four parts: routing, parking, matching and servicing.

Research on multi-agent systems is often (with good reason) done on simplified problems. For example, the shunting of trains is similar to Multi-Agent Path Finding, which is typically studied in a grid world [8]. The motivation for this work is twofold. First, the translation of a real-world problem to a conceptual model while making assumptions that are not preventing practical use is often a lot of work and requires frequent interaction with practitioners. We hereby aim to accommodate researchers in the multi-agent community with a simulator that encapsulates the most important practical details. Second, this simulator facilitates comparing different methods and developing new algorithms to solve the practical planning problem of train unit shunting and servicing.

TORS (TreinOnderhoud- en RangeerSimulator) is a sequential event-based simulator. It keeps track of the state that the shunting yards and the trains are in. In TORS, users can choose actions such as routing a train from one track to another or cleaning it. TORS then validates the action and updates the state according to the user (algorithm) input.

1.1 Related Work

A first mathematical model for the train unit shunting problem (TUSP) (without considering the service tasks) splits the problem

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

into a parking track assignment problem and a matching problem [1]. With this approach some real-life instances of the NS could be solved. Later, this formulation was extended to also include the routing of the trains through the shunting yard [6]. Both [4] and [5] provide an integrated approach for the two matching and parking sub-problems. A comparison benchmark of multiple solution approaches for TUSP, based on instances of the Danish State Railway and the NS is provided in [3]. They introduce three new methods: a constraint programming formulation, a column generation approach, and a randomized greedy heuristic.

TUSS extends the TUSP by including service tasks, such as cleaning, regular checks, and maintenance. For practical applications the TUSS problem can currently only be solved by heuristics [2]. A simulated annealing algorithm is presented in [10], which is the first algorithm to solve the complete problem (matching, combining and splitting, routing, parking, service scheduling) for real-world instances and is in the process of being adopted by Dutch Railways. A combination of machine learning with local search with the goal to provide bounds of the logistic hub capacity is presented in [9]. More recently [7] suggests a road map for a Multi-Agent Path Finding based model for TUSS.

2 PROBLEM DESCRIPTION

Trains are a combination of linked train units, which can in turn be seen as elementary agents. Routing the train units requires planners to find a conflict-free path for trains, starting when a train enters the yard, up to its departure in the morning. During this route, the train units should be parked on the yard. Extra complexity comes from the possibility to couple and decouple trains into combined units that require only one train operator. Additionally, trains are not easily reversed. To reverse a train, the train driver has to walk to the other side of the train and operate the train from there.

In most countries, the time table does not assign specific train units but rather trains of a specific type. This gives rise to a perfect matching problem on the bipartite graph where on one side we have a node for each train unit and on the other side we have the departure times for train units of certain types. Here, edges only exist between nodes if the train unit has the same type as is required by the departure. Trains of different types can also have different lengths and different speeds.

The servicing sub-problem considers that each train unit has a specified set of tasks, consisting of cleaning (inside and outside), regular maintenance checks and repairs. Such tasks can only be performed at certain locations on the shunting yard, such as a special washing track or a workshop.

The yard resembles a flexible flow shop [13], where physical trains are represented by the jobs and the train's maintenance tasks by the operations. In that setting, a set of tracks where an operation can be performed would be a set of similar machines. Due to the routing constraints however, this problem is more restricted than a general flexible flow shop. The informal goal of the TUSS problem is to specify a schedule for all servicing activities to be completed for each train in the provided interval that it is on the yard, finding collision free routes for each train to and from tracks where trains are parked and serviced, while respecting all kinds of safety "business rules" that were agreed upon with the rail infrastructure operator (ProRail in The Netherlands).

3 PRACTICAL DETAILS

To enable fast simulation, the core of TORS is written in C++. A python interface is provided to simplify development and testing of new algorithms. Specifically, the OpenAI Gym interface is implemented to comply to the reinforcement learning paradigm. The code is made open-source and available as a git repository [12].

TORS reads location and scenario data from files. With the configuration files it is easy to configure the level of realism that is desired in the simulation. The simulator can test a solution model's quality by running the simulation multiple times for generated scenarios with an increasing number of trains. The quality is determined by the number of trains it can find feasible plans for.

The simulation of TORS is state and action based. TORS provides the user with a state description and asks for an action. The state consists a.o. of the positions of the train units, current train combinations, the pending maintenance tasks and the arrival and departure times of trains.

4 CONCLUSIONS AND FUTURE WORK

In summary, the TORS simulator offers two main contributions. First, TORS facilitates researchers in multi-agent research in their work on the real-world Train Unit Shunting and Servicing problem. It does this by keeping track of the state of all agents, providing the user with a set of actions that correspond to the global state and updates the state according to the chosen actions. This facilitates both the use of anything from graph-based algorithms and operations research methods to reinforcement learning. TORS also evaluates the final plan that the user provides sequentially, enabling the comparison of different methods. TORS does all this by considering the routing, parking, matching and servicing sub-problems of TUSS.

To improve the link with reality even further we will include a few more aspects. First, the planning of personnel: train operators, cleaning crews and maintenance engineers each have their own set of skills and have to be scheduled to perform the planned tasks in practice. This addition includes much of the costs that are made in the real-world planning and could lead to better optimization criteria. Second, we will introduce uncertainty. There are several different events that disturb railway operations on a daily basis. Trains might not arrive at the specified time or could have unexpected maintenance or repair tasks. The duration of service tasks varies in practice. When combining these two extensions of the model, uncertainty of crew availability and walking times can also be taken into account, and a solution that performs well in the simulator will be directly useful useful in practice.

ACKNOWLEDGMENTS

This work has been developed in collaboration with the Software and Game project of the bachelor Computer Science in the faculty of Science, Utrecht University. The student team consisted of Dennis Arets, Sjoerd Crooijmans, Richard Dirven, Luuk Glorie, Jonathan den Herder, Jens Heuseveldt, Thijs van der Horst, Hanno Ottens, Loriana Pascual, Marco van de Weerthof, Kasper Zwijsen and was done under (NS) supervision by Joris den Ouden and Demian de Ruijter.

REFERENCES

[1] Richard Freling, Ramon M. Lentink, Leo G. Kroon, and Dennis Huisman. 2005. Shunting of passenger train units in a railway station. *Transportation Science* 39, 2 (2005), 261–272. https://doi.org/10.1287/trsc.1030.0076

[2] Martin Josef Geiger, Sandra Huber, Sebastian Langton, Marius Leschik, Christian Lindorf, and Ulrich Tüshaus. 2018. Multi-attribute assignment of trains to departures in rolling stock management: A contribution to the EURO/ROADEF 2014 Challenge. *Annals of Operations Research* 271, 2 (2018), 1131–1163. https://doi.org/10.1007/s10479-017-2501-8

[3] Jørgen Thorlund Haahr, Richard M. Lusby, and Joris Camiel Wagenaar. 2017. Optimization methods for the Train Unit Shunting Problem. *European Journal of Operational Research* 262, 3 (2017), 981–995. https://doi.org/10.1016/j.ejor.2017.03.068

[4] Leo G. Kroon, Ramon M. Lentink, and Alexander Schrijver. 2008. Shunting of passenger train units: An integrated approach. *Transportation Science* 42, 4 (2008), 436–449. https://doi.org/10.1287/trsc.1080.0243

[5] Ramon M Lentink. 2006. *Algorithmic Decision Support for Shunt Planning.* Ph.D. Dissertation. Erasmus Universiteit Rotterdam. http://hdl.handle.net/1765/7328

[6] Ramon M. Lentink, Pieter Jan Fioole, Leo G. Kroon, and Cor Van't Woudt. 2006. Applying Operations Research Techniques to Planning of Train Shunting. In *Planning in Intelligent Systems: Aspects, Motivations, and Methods.* John Wiley and Sons, 415–436. https://doi.org/10.1002/0471781266.ch15

[7] Jesse Mulderij, Bob Huisman, Denise Tönissen, Koos van der Linden, and Mathijs de Weerdt. 2020. Train Unit Shunting and Servicing: a Real-Life Application of Multi-Agent Path Finding. *arXiv preprint arXiv:2006.10422* (2020).

[8] Roni Stern, Nathan Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne T. Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, T. K. Satish Kumar, Eli Boyarski, and Roman Barták. 2019. Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks. *CoRR* abs/1906.08291 (2019). arXiv:1906.08291 http://arxiv.org/abs/1906.08291

[9] Arno Van De Ven, Yingqian Zhang, Wan Jui Lee, Rik Eshuis, and Anna Wilbik. 2019. Determining capacity of shunting yards by combining graph classification with local search. *ICAART 2019 - Proceedings of the 11th International Conference on Agents and Artificial Intelligence* 2 (2019), 285–293. https://doi.org/10.5220/0007398502850293

[10] Roel Van Den Broek. 2016. Train Shunting and Service Scheduling: an integrated local search approach. *Master thesis* (2016). https://doi.org/10.1007/s00105-004-0751-2

[11] R.W. van den Broek, J.A. Hoogeveen, J.M. van den Akker, and B. Huisman. 2021. *A Local Search Algorithm for Train Unit Shunting with Service Scheduling.* Technical Report. Utrecht University.

[12] Koos van der Linden, Mathijs M. de Weerdt, Bob Huisman, Joris den Ouden, Demian de Ruijter, J.A. Hoogeveen, J.M. van den Akker, Dennis Arets, Sjoerd Crooijmans, Richard Dirven, Luuk Glorie, Jonathan den Herder, Jens Heuseveldt, Thijs van der Horst, Hanno Ottens, Loriana Pascual, Marco van de Weerthof, and Kasper Zwijsen. 2021. cTORS. https://github.com/AlgTUDelft/cTORS.

[13] M Yazdani, M Amiri, and M. Zandieh. 2010. Flexible job-shop scheduling with parallel variable neighborhood search algorithm. *Expert Systems with Applications* 37, 1 (2010), 678–687. https://doi.org/10.1016/j.eswa.2009.06.007