Applying the Pebble Motion problem: studying the feasibility of the Train Unit Shunting Problem*

Issa K. Hanou, *Supervisors:* M.M. de Weerdt, and J. Mulderij

Delft University of Technology, Mekelweg 5, 2628 CD Delft, The Netherlands
i.k.hanou@tudelft.nl

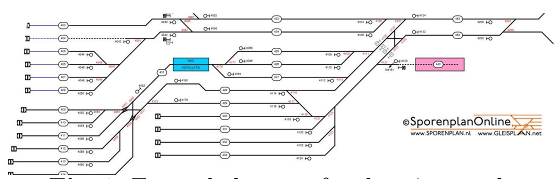

Fig. 1. Example layout of a shunting yard

Introduction The railways in the Netherlands are operated by the NS, *Nederlandse Spoorwegen*. As rail transportation offers a sustainable mode of transport, passenger flows have increased in the last decade and the NS strives to continue this development [6,7]. To support this trend, more trains are needed and the planning becomes more complicated. The problem we look at in this thesis considers the parking of trains in shunting yards (Figure 1) outside peak hours and is known as the (NP-hard) Train Unit Shunting Problem (TUSP) [3,5]. The NS currently uses a local search method to solve this problem [2], however, this algorithm is focused on finding a good solution and does not take the feasibility of a scenario into account. Therefore, the algorithm might end up using long computation times on scenarios that are infeasible to begin with.

In this thesis, we study different approaches to determine the feasibility of TUSP scenarios. We model TUSP as the Pebble Motion problem (PM), which considers the movement of pebbles in a graph, because PM scenarios are decidable in linear time [1]. The main differences between the TUSP and PM are i) the arrival and departure of trains at a shunting yard versus the pebbles already present in the graph; ii) the absolute arrival and departure times vs. time steps for movements; iii) the capacity of shunting tracks and trains lengths vs. unit-size pebbles and nodes; and iv) the train type composition vs. unique pebbles.

A restriction of the PM problem is created, where the graph is assumed to be a tree with a simple path at the root, and reallocation of pebbles is prohibited. This simple path models the arrival and departure sequences of trains in the shunting yard represented by the tree. These sequences resolve the differences i) and ii) and we call this problem variant the Pebble Motion on a Tree with Arrival and Departure (PMTAD). We first study this problem and later derive two extensions of this problem which respectively resolve differences iii) and iv).

To match the branch set of a tree, which expresses the size of each branch that can be reached from the root node, to the pebble sequences, we define a

* The full thesis can be found at http://resolver.tudelft.nl/uuid:2ca12cd2-315e-4a31-9a2a-67a77d8988bf

partition of a sequence. This partition is a set where each item is a totally ordered set (toset) such that all tosets are disjoint, the union of the tosets includes all pebbles, and the tosets respect the order of the pebbles in both the arrival and departure sequence. We prove that a partition and a branch set match means the PMTAD scenario that constructed them is a feasible scenario.

Studying the PMTAD First, we consider different scenarios and establish several conditions to determine the (in)feasibility of a problem instance based on the properties of the arrival and departure sequences, combined with the tree structure. We can derive bounds on the minimal number of required branches in the tree using a Directed Acyclic Graph representation of the two sequences.

Second, we assume problem instances have a given branch set definition. We refer to this restriction of the PMTAD as the Branch Set for a Partition Problem (BSPP). Studying this problem, we found an approach that runs in polynomial time and can find a partition to match the branch set. However, the main drawback is the fixed branch set. Since a tree can be expressed as one large branch or many smaller branches, as well as anything in between, this limits the possible partitions to match. An infeasible BSPP instance does not necessarily mean that the PMTAD instance from which it originated is also infeasible.

Finally, we consider instances of the PMTAD where the partition is given and we want to match it to a given shunting tree. We refer to this problem as the Partition for a Pebble Sequence on a Tree problem (PPST). We found an algorithm with a worst-case runtime that is exponential in the number of pebbles. Furthermore, we identified a special case of this problem with m branches in the tree, with at least two nodes each, and $n \leq 2m$ pebbles. This special case can be solved in polynomial time, as such an instance can be converted to a Bipartite Matching problem, which is in the complexity class P [8].

Extending the PMTAD Next, we create an extension of the PMTAD that also includes the length of trains and tracks: the Pebble Motion on a Tree with Arrival, Departure, and Length inclusion (PMTADL). Here, the pebbles are assigned a size and the branches in the associated branch sets are given a size, too. We consider the PPSTL extension of the PPST that also includes these sizes and show that this problem is NP-hard by creating a reduction of the classic Partition problem [4, SP12] to the PPSTL. Furthermore, we show that PMTADL and the PPSTL are closely related, and (in)feasible instances of one can be mapped to (in)feasible instances of the other in polynomial time.

Finally, we establish a different extension of the PMTAD that does not include the lengths but considers pebbles to be of a certain type, or color, where only the types of the pebbles have to be correctly matched between the arrival and departure sequences. A special case is considered where there are as many branches in the tree as there are different types considered in the problem, and we prove this is decidable in polyonimal time.

Conclusion We propose a PM variant which is more representative of transportation problems like the TUSP because the PMTAD is relevant for scenarios where sequences of vehicles are considered that must be parked or permuted in a limited space. Two extensions of the problem are discussed that further resemble real-world scenarios and the PMTADL is shown to be NP-complete.

References

1. Auletta, V., Monti, A., Parente, M., Persiano, P.: A linear-time algorithm for the feasibility of pebble motion on trees. Algorithmica **23**, 223–245 (1999)
2. van den Broek, R., Hoogeveen, H., van den Akker, M., Huisman, B.: A local search algorithm for train unit shunting with service scheduling. Transportation Science (2021)
3. Freling, R., Lentink, R.M., Kroon, L.G., Huisman, D.: Shunting of passenger train units in a railway station. Transportation Science **39**(2), 261–272 (2005)
4. Garey, M.R., Johnson, D.S.: Computers and Intractability, A Guide to the Theory of NP-Completeness. Bell Telephone Laboratories, Incorporated, United States of America (1979)
5. Lentink, R.: Algorithmic decision support for shunt planning. Phd dissertation, Erasmus Universiteit Rotterdam (2006)
6. NS: Jaarverslag 2019. Report, *Nederlanse Spoorwegen* (2019)
7. NS: Vervoerplan 2022. Report, *Nederlanse Spoorwegen* (2022)
8. Sedgewick, R.: Algorithms in Java, Part 5 graph algorithm. Addison-Wesley Pearson Education, 3rd edn. (2004)