Hump Yard Sorting in one Humping Step - Minimizing the Number of Back-and-Forth Movements to form the Outbound Sequence

Elias Dahlhaus[1], Luka V. Sabljic[2]

[1] Department of Computer Science, Darmstadt University of Technology, 64289 Darmstadt, Germany
dahlhaus@algo.informatik.tu-darmstadt.de

[2] Department of Electrical Engineering and Information Technology, Darmstadt University of Technology, 64283 Darmstadt, Germany
Luka91@gmx.de

Abstract

For the last mile, it is reasonable to sort freight cars in a fixed order. Since coupling of cars is time consuming, it should be avoided to push any car more than once through the hump. In this paper, an algorithm is proposed that distributes the cars into the classification tracks, such that in each track, the cars form a monotonic subsequence with respect to the outbound order. This makes it possible tor form the outbound train with a certain number of back-and-forth movements of the locomotive. The algorithm takes care that the number of back-and-forth movements is minimized. The number of classification tracks that are available is part of the input.

Keywords : *Hump yard, permutation, graph coloring, minimum cost flow.*

1 Introduction

Since coupling of cars is time consuming and drawing cars back through the hump would prevent other sorting processes, one is interested in sorting the cars of a certain car sequence in one humping step where the cars are pushed through the hump to the classification tracks. It should also be the case when the cars should be sorted in a fixed order as it is for last mile distributer trains. That means we are given a permutation that maps, for each car, its inbound position to its outbound position. We can proceed as follows. 1. We determine the (inclusion maximal) intervals of outbound positions, such that the permutation is monotonically increasing, called chains. 2. For each chain, we provide a classification track. The cars belonging to the ith chain are put into the i-th classification track. 3. Finally the chains are concatenated to the sorted sequence. This is more or less known, see for example [1]. The number of available classification tracks might be less than the number of chains. 2 R. Hansmann followed the following idea in his Doctoral dissertation [2]. One considers two chains as not being in conflict if the inbound positions of all wagons of the first chain are less than the inbound positions of all wagons of the second chain and the outbound positions of all wagons of the first chain are less than the outbound positions of all wagons of the second chain. Chains not being in conflict can be put into the same track. In his doctoral thesis, R. Hansmann reduced the problem to minimize the number of tracks to the problem to color trapezoid graphs.. That can be solved efficiently [4]. Next one can consider the problem that the number of available tracks is still not large enough. One could get the idea to split the permutation into a minimum number of monotonically increasing subsequences. But then the number of back-and-forth movements to form the final sequence might be quite large. It can be as large as the number of cars. The contribution of this paper is the following. It is given a sorting requirement by a permutation and the number of tracks. We assume that the number of tracks is less than the number tracks that are needed to sort the cars in such a way that the chains are kept as a whole as in the doctoral thesis of [2]. It is to find as many monotonically increasing subsequences of the permutation as the given number of available tracks, such that as few chains are broken as possible. The number of subchains appearing in the classification tracks is a measure, how many back-and-forth movements are necessary.

2 Basic Approach

It is assumed that, for the given permutation of wagons, a conflict free distribution of the chains to classification tracks has been done and that the number of classification tracks is minimal. One considers the cars as nodes of a directed graph. One provides a blue directed edge from a wagon x to a wagon y if x and y belong to the same classification track and y is the successor wagon of x. Next another distribution of the wagons to the classification tracks is considered, such that k tracks less are needed. We consider a directed graph $D = (V, A)$ with the car set as vertex set V and an arc set A that consists of *blue* and *read* arcs.

(x, y) is a blue arc if if they are in the same track of the first distribution and y is the successor car of x. Analogously if x and y belong to the same track in the second distribution and y is the successor car of x then (x, y) is called a read edge.

Note that the blue edges form a disjoint of directed paths and the read edges form a disjoint union of k less paths.

The edges that are read or blue but not both form an edge-disjoint union of read-blue alternating paths where read edges are passed in reverse direction. All these paths start at vertices having only a leaving blue edge with a read edge as its first edge. They are exactly k many paths.

It is to find a collection of k such paths, such that the number blue arcs passed by these paths is minimized.

3 Minimum Cost Flow Approach

The directed graph is now defined as follows.

1. For each car x, we provide an *upper vertex* v_x and a *lower vertex* w_x.
2. If (x, y) is a blue edge then (w_x, v_y) forms an arc of cost 1, if x and y have consecutive inbound positions. Otherwise, (w_x, v_y) is of cost 0.
3. If x has a smaller inbound position that y and a smaller outbound position than y than an arc (v_y, w_x) is provided (reverse of a possible read arc). These arcs are of cost 0.
4. A source node S is provides with arcs to all v_x, such that x is the first wagon of a path of blue edges on the cars, and a target node T is provided with arcs from all w_y, such that y is the last car of a path of blue edges. These arcs are also of cost 0.

As explained also in [3], we apply the known successive shorstest path algorithm to this weighted graph. to find k paths. This can be transformed into k less read paths and therefore a distribution of the cars into k less tracks.

4 Conclusion and perspectives

The case that the length of classification tracks is bounded has not been considered. It is to expect that an exact solution is NP-complete. On the other hand, in many hump yards, the length of classification tracks exceed 700 m . Since in Europe, the train lengths usually do not exceed 750 m, the length restrictions of classification tracks do not come in consideration in most cases.

5 Acknowledgement

The implementation of the algorithm is part of the Bachelor Thesis of the second author [5].

References

[1] M. Miller J. Ryan E. Dahlhaus, P. Horak. The Train Marshalling Problem. *Discrete Applied Mathematicsl*, 103(1-3):. 41–54, 2000.

[2] R. Hansmann. *, Optimal Sorting of Rolling Stock.* Doctoral Dissertation, Braunschweig University of Technology and Cuvillier Publishing Company, GÃ¶ttingen, 2011.

[3] J.B Orliin R.K. Ahuja, T.L Magnanti. *, Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, New York, 1993.

[4] L. Wernisch S. Felsner, R. MÃ¼ller. Trapezoid Graphs and Generalizations, Geometry and Algorithms. *Discrete Applied Mathematicsl*, 74(1):. 13–32, 1997.

[5] L. Sabljic. *, Coloring Permutations with Constraints and their Application in Efficient Sorting in Marshalling Yards.* Bachelor Thesis, Department of Electrical Engineering and Information Technology, Darmstadt University of Technology, 2017.