Train Marshalling is Fixed Parameter Tractable

Leo Brueggeman[1], Michael Fellows[2], Rudolf Fleischer[3]*, Martin Lackner[4]**, Christian Komusiewicz[5], Yiannis Koutis[6], Andreas Pfandler[4]**, and Frances Rosamond[2]

[1] University of California, Santa Cruz, USA, Email: l.a.brueggeman@gmail.com
[2] Charles Darwin University, Darwin, Australia,
Email: michael.fellows@cdu.edu.au, frances.rosamond@cdu.edu.au
[3] SCS and IIPL, Fudan University, Shanghai, China, and GUtech, Muscat, Oman
Email: rudolf.fleischer@gutech.edu.om
[4] Vienna University of Technology, Austria,
Email: lackner@dbai.tuwien.ac.at, pfandler@dbai.tuwien.ac.at
[5] TU Berlin, Germany, Email: christian.komusiewicz@tu-berlin.de
[6] University of Puerto Rico, Email: i.koutis@gmail.com

Abstract. The train marshalling problem is about reordering the cars of a train using as few auxiliary rails as possible. The problem is known to be NP-complete. We show that it is fixed parameter tractable (FPT) with the number of auxiliary rails as parameter.

1 Introduction

Consider the following railway problem. If a train splits into several trains going to different destinations, the cars must be ordered such that the first few cars have destination 1, the next few cars have destination 2, etc. If an incoming train has the cars unordered, we must rearrange the cars. The only way to rearrange cars is to bring the train to a shunting yard where the rail splits into several auxiliary rails. Then the cars of the train are decoupled and moved, one by one, to the auxiliary rails where they form smaller trains. Finally, the small trains from the auxiliary rails are again connected to form a new train that is properly ordered, i.e., the first few cars have destination 1, the next few cars have destination 2, etc. The goal is to reorder the cars using as few auxiliary rails as possible (because a train station has only limited number of such rails, and they might be needed to sort several trains simultaneously).

* This work was supported by a grant from the NSFC (No. 60973026), the Shanghai Leading Academic Discipline Project (no. B114), and the Shanghai Committee of Science and Technology (no. 08DZ2271800).

** Supported by the Austrian Science Fund (FWF), project P20704-N18.

We can reformulate this problem as a problem of rearranging sequences. Assume we have a partition of the set $\{0, 1, \ldots, n-1\}$. The numbers are the car numbers of the incoming train, and each set of the partition corresponds to a destination. We now want to map the partition sets into $(0, 1, \ldots, n-1, 0, 1, \ldots, n-1, \ldots, 0, 1, \ldots, n-1)$, where we have k repetitions of $(0, 1, \ldots, n-1)$. The goal is to minimize k while mapping the sets such that all elements of the first set are mapped before all elements of the second set, etc. We call this the TRAIN MARSHALLING PROBLEM (TM).

Example 1. Assume the train consists of 11 cars. There are 6 destinations: A, B, C, D, E and F. Therefore A to F can be considered as a partition of the set $\{1, \ldots, 11\}$. Let $A = \{2, 4\}$, $B = \{6, 8\}$, $C = \{3, 7, 10\}$, $D = \{9\}$, $E = \{1, 11\}$ and $F = \{5\}$. This train can be reordered with 3 auxiliary rails: The first rail contains (starting with the first car) $2, 4, 6, 8, 10$ (destinations A, B, C), the second rail $3, 7, 9, 11$ (destinations C, D, E) and the third rail $1, 5$ (destinations E and F). Coupling the trains starting with the first rail and followed by the second and third rail yields the following order:

$$\underbrace{2, 4}_{A}, \underbrace{6, 8}_{B}, \underbrace{10, 3, 7}_{C}, \underbrace{\ 9\ }_{D}, \underbrace{11, 1}_{E}, \underbrace{\ 5\ }_{F}.$$

According to Dahlhaus et al. [3], the problem was first mentioned in an old paper published in China [1]. Later, Zhu and Zhu [4] gave polynomial time solutions for some special cases of the problem. Dalhaus et al. then showed that the problem is NP-hard by reduction from NUMERICAL MATCHING WITH TARGET SUMS (NMTS) [3] and gave algorithms for related problems [2]. Their NP-completeness reduction uses a construction with $k = \Theta(n)$, so we may ask whether TM is fixed parameter tractable when parameterized by k. In this paper we give an affirmative answer to this question. We show that we can indeed compute a solution in time $O(2^{O(k)} \cdot poly(n))$. Our algorithm is based on a clever dynamic program to compute the minimum density of a mapping, a concept that is closely related to the original k-TM problem.

This paper is organized as follows. After defining the problem and discussing some basic properties in Section 2, we propose a dynamic program for k-TM in Section 3. We end the paper with concluding remarks in Section 4.

2 Preliminaries

Let $[n] = \{0, \ldots, n-1\}$ denote the set of integers from 0 to $n-1$. A partition $S = \{S_0, \ldots, S_{t-1}\}$ of $[n]$ is a collection of non-empty and pairwise disjoint subsets S_i of $[n]$ such that $\bigcup_i S_i = [n]$. Let $K(S)$ be the smallest number k such that there exists a permutation $\pi = (\pi_1, \ldots, \pi_t)$ of $[t]$ such that we can map the sets of the partition into $[n]^k = (0, 1, \ldots, n-1, 0, \ldots, n-1, 0, \ldots, n-1)$, where $0, \ldots, n-1$ is repeated k times (we call each repetition a *phase*), in such a way that all elements of S_{π_i} appear before all elements of S_{π_j}, for all $i < j$. We call such a mapping a TM-mapping. The Train Marshalling Problem (TM) is then defined as follows.

Problem Train Marshalling Parameterized by k (k-TM)
Input: A partition S of $[n]$ and $k \geq 1$;
Parameter: k;
Question: Is $K(S) \leq k$?

We may assume w.l.o.g. that all sets S_i have size at least two. This can easily be achieved by replacing each set $\{x\}$ by a size-two set $\{x, x + \frac{1}{2}\}$. This is because such sets would always be scheduled in the order $\{x, x + \frac{1}{2}\}$ in an optimal solution, i.e., without a wrap-around (see below). For example, the train in Example 1 would become Let $A = \{2, 4\}$, $B = \{6, 8\}$, $C = \{3, 7, 10\}$, $D = \{9, 9.5\}$, $E = \{1, 11\}$ and $F = \{5, 5.5\}$.

A set $S_i = \{a_1, \ldots, a_m\}$ of the partition with $a_1 < \cdots < a_m$ induces m half-open intervals $[a_1, a_2), [a_2, a_3), \ldots, [a_{m-1}, a_m), [a_m, a_1)$, called *segments*, on the circle $(0, 1, \ldots, n-1, n = 0)$, where we identify n with 0. We call $[a_m, a_1)$ the *wrap-around* segment (it is the only segment containing the interval $[n-1, 0)$), the other segments are called *proper*. The union of all proper segments of S_i forms the half-open interval $I_i = [a_1, a_m)$ which we call the *proper interval*. Note that no two proper intervals share a common endpoint because each proper interval corresponds to a set of the partition of S.

Let $b \in [0, n)$. Then b is contained in exactly one segment of each set of the partition. We denote by $cw(b)$ the number of proper segments containing b. The *cut-width* of S is then defined as $cw(S) = \max_b cw(b)$. Intuitively, $cw(S)$ is the cut-width (or clique number) of the interval graph induced by the proper intervals I_i, for $i = 0, \ldots, t-1$.

We may w.l.o.g. assume that an optimal TM-mapping maps the numbers in S_i in consecutive order into $[n]^k$, with a possible wrap-around from

$n-1$ to 0, i.e., they appear in the order $a_j, \ldots, a_m, a_1, \ldots, a_{j-1}$, for some $j = 1, \ldots, m$. For example, the set $\{2,5,7\}$ may be mapped as $(2,5,7)$, $(5,7,2)$, or $(7,2,5)$. We call the segment $[a_{j-1}, a_j)$ *passive*, the other segments *active*. In the example above, the mapping $(5,7,2)$ would render the segments $[5,7)$ and $[7,2)$ active while the segment $[2,5)$ would be passive.

For a fixed TM-mapping and any number $b \in [0,n)$, let the *density at b*, denoted by $d(b)$, be the number of active segments containing b. We note that $d(b)$ depends on the TM-mapping, but it should always be clear from the context which TM-mapping is meant. We say the partition S has *k-bounded density* if there is a TM-mapping such that $d(b) \leq k$ for all $b \in [0, n-1)$ and $d(b) \leq k-1$ for all $b \in [n-1, 0)$. The *proper density* is defined similarly except that we only count active proper segments. It turns out that solving k-TM is equivalent to finding a TM-mapping of k-bounded density.

Theorem 2. *A partition S of $[n]$ is a* Yes*-instance of k-*TM *if and only if S has k-bounded density.*

Proof. If we have a TM-mapping of S into $[n]^k$, then clearly $d(b) \leq k$ for $b \in [0, n-1)$ (sets are mapped overlap-free into each of the k phases of $[n]^k$), and $d(b) \leq k-1$ for $b \in [n-1, 0)$ (there can be at most $k-1$ wrap-arounds).

On the other hand, if the density is k-bounded, then we can greedily compute a TM-mapping of S into $[n]^k$. We think of the k phases of $[n]^k$ as the k layers of an interval graph of clique number k. We first split at 0 all wrap-around sets into two intervals, then we sort all intervals (the two split intervals for wrap-around sets, and the intervals I_i for non-wrap-around sets) by left endpoint and assign them greedily from left to right to a free layer. The wrap-around segments then tell us in which order to map the layers to the k phases. □

3 An FPT Algorithm for k-TM

Our FPT algorithm for k-TM tries to choose one segment in each partition set of S as the passive segment such that the density is k-bounded everywhere. By Theorem 2, this solves k-TM on S. The algorithm takes advantage of the close relationship between the parameter k and the cutwidth $cw(S)$ of S.

Lemma 3. *If a partition S of $[n]$ is a* Yes*-instance of k-*TM*, then $cw(S) \leq 2k-1$.* □

Proof. If $cw(S) \geq 2k$, then there exists $b \in [0, n-1)$ such that $cw(b) \geq 2k$. But then either $d(b) \geq k+1$ or $d(c) \geq k$ for all $c \in [n-1, 0)$ (i.e., we have at least k wrap-arounds), which is impossible by Theorem 2 for YES-instances of k-TM. □

This means, we only need to solve k-TM on instances of cut-width at most $2k-1$ (we immediately say NO in all other cases). We now describe a dynamic program that runs efficiently on partitions of bounded cut-width. The density of an interval $[i, j)$ depends on which segments overlap with the interval and which of them are passive. To compute the density with a dynamic program, we must keep track of the segments that are cut at the boundaries i and j. There can be at most $cw(S)$ proper segments in each cut and at most $cw(S)$ wrap-around segments of partition sets where the other cut intersects a proper segment, but there might be many wrap-around segments that intersect both cuts. However, we only need to count their number, and this is the key to build a small dynamic programming table. Also, it turns out to be more convenient to compute proper densities instead of densities.

To be more precise, we construct a four-dimensional table T to compute proper densities. The first two dimensions, with indices i and j, respectively, specify a range $[i, j)$, where $0 \leq i < j \leq n-1$. The third dimension $\ell = 0, \ldots, k$ denotes the maximum density in $[i, j)$ we want to achieve. The fourth dimension represents the two cuts at i and j. Any such cut will hit at most $cw(S)$ proper segments of S. A proper segment may be either active or passive. We must also distinguish whether the corresponding partition set has its passive interval overlapping with $[i, j)$ or not. We collect all this information in a *boundary configuration*, denoted by $C_{i,j}$. To be more precise, $C_{i,j}$ contains for each partition set with proper segments intersecting $[i, j)$ the following information: Whether the segment containing i is active or passive, whether the segment containing j is active or passive (it could be the same segment as the segment containing i), and whether one of the segments of the partition intersecting $[i, j)$ is passive. We say two boundary configurations $C_{i,t}$ and $C_{t,j}$ are *consistent* if they could both occur simultaneously for some TM-mapping (e.g.., they do not define two passive segments, and they classify the segment containing t the same way). Note that the number of boundary configurations is at most $2^{4cw(S)}$.

Some partition sets may have their proper intervals contained in $[i, j)$. In this case, the corresponding wrap-around interval contains $[j, i)$. If that segment is active, it contributes to the density (but not the proper density) of any interval in $[j, i)$. We therefore store in $T[i, j, \ell, C_{i,j}]$ the minimum number of active wrap-around segments containing $[j, i)$ under the constraints specified by the boundary configuration $C_{i,j}$ and such that the proper density in $[i, j)$ is at most ℓ. Note that the density in $[i, j)$ can then be computed as the proper density in $[i, j)$ plus the $T[.]$ values of $[0, i-1)$ and $[j+1, n-1)$ (note that a wrap-around segment can only be counted for one of the two disjoint ranges).

We can compute the table T recursively as follows. We first describe the base case. Let $i \in \{0, \dots, n-2\}$, $\ell \in \{0, \dots, k\}$, and $C_{i,i+1}$ a boundary configuration. If the cut at i intersects more than ℓ active proper segments, this boundary configuration cannot achieve proper density at most ℓ in $[i, i+1)$, so we set $T[i, i+1, \ell, C_{i,i+1}] = \infty$. Otherwise, we set it either to 0 or 1, depending on whether the segment $[i, i+1)$ is active or passive, respectively. Note that the partition set $\{i, i+1\}$ is the only one that could induce a wrap-around segment that needs to be counted in $T[.]$. Since an optimal solution would always choose $[i, i+1)$ as the active segment, we could also set $T[i, i+1, \ell, C_{i,i+1}] = \infty$ for configurations where $[i, i+1)$ is a passive segment.

To compute the value for larger ranges, let $i < t < j$ and let $C_{i,t}$, and $C_{t,j}$ be consistent boundary configurations. Any active wrap-around segment counted in either $T[i, t, .]$ or $T[t, j, .]$ should be counted in $T[i, j, .]$. But there might also be additional segments to be counted if a wrap-around segment contains $[j, i)$ but none of the two smaller ranges and if this segment is active. This information can be obtained from the two boundary configurations. Let s denote the number of these segments. If we have proper density ℓ_1 in $[i, t)$ and proper density ℓ_2 in $[t, j)$, then we can achieve proper density

$$d(\ell_1, \ell_2) = \max\{\ell_1 + T[t, j, \ell_2, C_{t,j}], \ell_2 + T[i, t, \ell_1, C_{i,t}]\}$$

in $[i, j)$ with

$$w(\ell_1, \ell_2) = T[i, t, \ell_1, C_{i,t}]\} + T[t, j, \ell_2, C_{t,j}] + s$$

active wrap-around segments containing $[j, i)$. Thus,

$$T[i, j, \ell, C_{i,j}] = \min_{d(\ell_1,\ell_2) \leq \ell} w(\ell_1, \ell_2)$$

This finishes the description of the algorithm.

Theorem 4. *The dynamic program solves* k-TM *in time* $O(2^{O(k)} \cdot poly(n))$, *i.e.,* k-TM *is fixed parameter tractable with parameter* k.

Proof. The size of the table is exponential in $cw(S)$ and thus exponential in k (by Lemma 3) but polynomial in n. Each table computation needs polynomial time. □

4 Conclusions

We have shown that we can use dynamic programming to solve the Train Marshalling Problem in time $O(2^{O(k)} \cdot poly(n))$, i.e., we have shown that the problem is fixed parameter tractable. It would be interesting to find a small kernel for the problem.

Acknowledgements

This paper is one of the results we obtained at the workshop *Parameterized Complexity: Not About Graphs!* in Darwin at CDU in August 2011 and the subsequent problem-solving bush workshop in the alligator-infested jungle around Darwin. These two workshops were great fun and very productive and we would like to thank CDU for the great support.

References

1. *Acta Mathematicae Applicatae Sinica*, 1(2):91–105, 1978.
2. E. Dalhaus, P. Horak, M. Miller, and J. F. Ryan. Algorithms for combinatorial problems related to train marshalling. *CiteSeerX*, (10.1.1.37.4090), 2000. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.37.4090.
3. E. Dalhaus, P. Horak, M. Miller, and J. F. Ryan. The train marshalling problem. *Discrete Applied Mathematics*, 103(1–3):41–54, 2000.
4. Y. Zhu and R. Zhu. Sequence reconstruction under some order-type constraints. *Scientia Sinica*, 26(7):702–713, 1983.