Delft University of Technology
Parallel and Distributed Systems Report Series

Complexity of Pre- and End-Haulage Problems

Leon Aronson    Cees Witteveen

report number PDS-2002-002

ISSN 1387-2109
Contents

1 Introduction 2

2 General problem description 4
  2.1 Variants ......................................................... 6
  2.1.1 Number of terminals ............................................ 6
  2.1.2 Number of trucks ............................................. 6
  2.1.3 Timing constraints ............................................ 6
  2.1.4 Infrastructural conditions ................................. 6

3 Result summary 7

4 Planning with one terminal and no time windows 10

5 Planning with one terminal, release times, and deadlines 13
  5.1 Symmetry and triangle inequality ............................. 15
  5.2 More time constraints ......................................... 15

6 Planning with one terminal, deadlines, and multiple trucks 16
  6.1 Symmetry and triangle inequality ............................. 17

7 Planning with multiple terminals 18
  7.1 Symmetry and triangle inequality ............................. 19

8 Conclusion and further research 20

A Time windows 21
Chapter 1

Introduction

This report discusses the complexity of Pre- and End-Haulage Problems (PEHPs), which occurs in intermodal drayage. It is not as much a contribution to the field of algorithmic complexity theory as it is an application of that theory to a specific kind of transport problems.

Globally, a PEHP is the problem of determining truck movements such that the transport demand is satisfied and the costs, which depend on time and distance, are minimized. The transport demand consists of orders specifying a source and a destination, one of which must be an intermodal terminal, and time constraints for pickup and delivery. Thus, freight must either be transported from terminal to customer (end-haulage), or from customer to terminal (pre-haulage). The challenge is to combine pre-hauls with end-hauls as much as possible, to minimize the cost related to 'empty' traveling. However, this can be complicated by time constraints. For example, suppose a truck has delivered a freight at a customer, and a pre-haul is not directly available. Should the truck then

- wait until a pre-haul is available at the current customer,
- drive to another customer to perform a pre-haul, or
- return to the terminal to perform another end-haul?

The decision depends on the cost function and the time constraints. It may be cheaper to wait, but if another haul is in a hurry it may be better to execute that haul immediately. The situation can be complicated even further in a dynamic situation, in which not all hauls are known in advance. However, we have restricted ourselves to static problems.

Most general transport optimization problems are NP-hard. However, the types of movements a vehicle can make in a PEHP are severely restricted. A vehicle either drives from a terminal to a customer and back, or from a terminal via a customer to another customer and then back to the terminal (combined haul). Due to this restriction it is interesting to ask which variants of this problem are NP-hard.
Although we are interested in the \textit{NP-hardness of optimization problems}, in this report a number of cases can be found in which we show the \textit{NP-completeness of the corresponding decision problems}. We trust that the reader can derive the desired result on the corresponding optimization problems.

Since we are developing a dynamic planner for PEHPs (see [1]), we need to know whether optimal solutions can be computed quickly. We do not know of any publications explicitly dealing with the complexity of PEHPs, therefore some research was needed to obtain the desired knowledge. This report is the result of that research. Furthermore, it seems that some researchers still are not aware of the inherent difficulty of different variants of the problem. For example, a static planner has been developed by Spasovic ([6]), but the planner absorbs too much time to be used in dynamic planning, besides still being nonoptimal. However, the issue of NP-hardness is not mentioned, although Spasovic states that further work has to be done to reduce computing time.

Our problem has similarities to other transport problems, mainly the Pickup and Delivery Problem (PDP) and the Vehicle Routing Problem (VRP). The NP-completeness of this problem is well-known and many algorithms have been designed to obtain good solutions (e.g. [4], [2]). Therefore we suspected that the PEHP is NP-complete, too. This report indeed shows that most variants of the PEHP, except for the simplest ones, are NP-complete. Despite of the similarities with PDPs, we mostly preferred to use scheduling problems (see [3]) as a basis for the reductions in this report.
Chapter 2

General problem description

We consider a transport problem occurring in intermodal transport. It is known as the pre- and end-haulage problem (PEHP). The name refers to the first and the final part of moving a freight; while the largest portion of the total distance the freight travels by rail, it has to travel two smaller parts by road to enable the rail transport.

Envision a (railroad) terminal where freights (usually containers) leave and arrive by train. Arriving freights are loaded on a truck for transport to their final destination (end-haulage). The leaving freights have been transported from their origin to the terminal by trucks (pre-haulage). The problem we are concerned with is how to use the available trucks such that all pre- and end-haulage tasks can be performed with minimal costs. The costs may consist of time costs, distance costs, and penalty costs (if time constraints are violated). The trucks are not necessarily bound to one terminal. It is possible that multiple terminals should be dealt with by the same fleet of trucks. In that case, we assume that for each freight the terminal where it should be delivered or picked up is predetermined.  

To describe the area considered in a pre- and end-haulage problem is based on a complete directed graph $G = (V, E)$. $V$ is a set of locations which can be partitioned into a set of railroad terminals $V_{term}$ and a set of customer locations $V_C$. The edges in the graph represent roads. Each road, going from location $x$ to location $y$, is labeled with a distance $\delta_{xy}$ and a travel time $\tau_{xy}$. We assume that these values are 0 if and only if $x = y$.

The problem to be solved is described, besides the infrastructure, as a set of orders $O$. Each order $o_i \in O$ consists of

- a pickup location $s_i$ and a delivery location $d_i$;
- pickup and delivery durations $w_{s_i}$ and $w_{d_i}$; and
- pickup and delivery time windows $T_{s_i}$ and $T_{d_i}$.

\footnote{The problem of choosing a terminal, given the destination of a freight, is outside the scope of this report.}
The definition of a time window and operations involving them can be found in Appendix A. Each order $o_i \in O$ is subject to the following restrictions:

- either the pickup or the delivery location is a railroad terminal; i.e. $s_i \in V_{term}$ or $d_i \in V_{term}$;
- the order can theoretically be delivered in time, that is, the sum of the earliest possible pickup time, the delivery duration, the travel time between pickup and delivery location, and the delivery duration yields a time which must fall before the latest possible delivery time;
- the lengths of the pickup and delivery time windows must be at least equal to the pickup and delivery durations, respectively: $|T_{s_i}| \geq w_{s_i}$ and $|T_{d_i}| \geq w_{d_i}$.

The last three restrictions provide that all orders are individually feasible. When appropriate, $O$ is partitioned into a set of pre-haulage orders $O_p$ and a set of end-haulage orders $O_e$. If $o_i \in O_p$, then $d_i \in V_{term}$; if $o_i \in O_e$, then $s_i \in V_{term}$. The set of available trucks is called $A$. Each potential solution to the problem can be described in the form of a (truck) schedule. A truck schedule for truck $a \in A$ is a sequence $\sigma(a)$ of truck moves; each move is described by a start location $\tau_{a,j}$, a destination location $\tilde{d}_{a,j}$, and a time window $T_{a,j} = [\tilde{t}_{a,j}^s, \tilde{t}_{a,j}^d]$, where $j$ is the number of a move, and $\tilde{t}_{a,j}^s, \tilde{t}_{a,j}^d \in \mathbb{Z}$ mark the moments in time between which the truck move is performed. We define $|\sigma(a)|$ to be the number of moves of truck $a$ under schedule $\sigma$. The following must hold for all relevant $a$ and $i$:

- $\tau_{a,j} = \tilde{d}_{a,(j-1)}$ if $j \geq 2$ (truck moves are continuous);
- $|T_{a,j}| \geq \tau_{a,j} \tau_{a,j}$ (time windows are sufficiently large); and
- $\tilde{t}_{a,j}^s \geq \tilde{t}_{a,(j-1)}^d$, if $j \geq 2$ (truck moves are consecutive, not simultaneous).

A quality measure of a truck schedule $\sigma(a)$ is the total distance traveled by the truck $a$ according to $\sigma(a)$.

Given the order set $O$, a solution is a truck schedule which serves all orders. This means that there is an injective function from orders to truck moves, such that the mapped truck move satisfies the order constraints. Thus, we an injective function $f : \{1, \ldots, |O|\} \rightarrow A \times \mathbb{Z}$ must exist such that order $o_i$ is mapped on truck move $j$ of truck $a$, where $f(i) = (a, j)$, and the following restrictions hold:

- $f(i) = (a, j)$ is an existing truck move: $1 \leq j \leq |\sigma(a)|$;
- the start location of the move is the pickup location of the order: $s_i = \tau_{f(i)}$;
- the pickup falls within the pickup time window: $T_{f(i)}^s \in T_{s_i}$ and $T_{f(i)}^s + w_{s_i} \in T_{s_i}$.
• the end location of the move is the delivery location of the order: \(d_i = \overline{d_{f(i)}}\);
• the delivery falls within the delivery time window: \(T_{f(i)}^l \in T_{d_i}\) and \(T_{f(i)}^l - w_{d_i} \in T_{d_i}\); and
• pickup, travel, and delivery can all be performed within the time window of the truck move: \(w_{s_i} + \tau_{s_i} d_i + w_{d_i} \leq |T_{f(i)}^l|\).

2.1 Variants

This report considers some variants of the PEHP. Each variant has an unique set of restrictions. For each variant we will try to answer the question whether it is in P or NPC. The restrictions which may be present in a variant are shortly discussed below.

2.1.1 Number of terminals

One possible simplification of the PEHP is the restriction to one terminal. This does not necessarily make the problem polynomially solvable, but, as we will see, the multiple-terminal problem is NP-complete.

2.1.2 Number of trucks

Another simplification of the PEHP is the restriction to one truck. For most situations, the number of trucks does not alter the membership of P or NPC.

2.1.3 Timing constraints

Timing constraints usually complicate matters. Although there are some scheduling problems with timing constraints which can be solved in polynomial time (e.g. see [3], comments at [SS1] and [SS8]), generally little timing is needed to enter NPC.

2.1.4 Infrastructural conditions

There are two graph conditions which are used a lot in all kinds of planning problems: symmetry and the triangle inequality. As we will see, both do not influence the complexity of pre- and end-haulage problems.
Chapter 3

Result summary

In this chapter we give a short overview of the results derived in this report. The results concern membership of P versus NPC of several variants of the general PEHP described in the previous chapter. These variants are determined by the presence or absence of the following restrictions: '1 terminal', 'no deadlines', '1 truck', and 'no release times'. Infrastructural properties like symmetry and the triangle inequality do not influence the complexity. The results have been summarized in a Hasse diagram (Figure 3.1). In the Hasse diagram, two variants of the problem are connected if they differ in exactly one restriction; the upper problem is the more general one. Thus, the problem at the top is the general PEHP, while the problem at the bottom is the simplest variant we consider. The 'dark' problems are in NPC, while the 'white' problems are in P. The complexity of the lightly-shaded problem is open.

From the diagram we draw the following conclusions. In the first place, the majority of the variants under consideration is in NPC, while the realistic degree of the P-complete variants is minimal.

Secondly, the restrictions '1 terminal' and 'no deadlines' are the main indicators of NPC-membership.

- '1 terminal': All variants with multiple terminals are in NPC.
- 'no deadlines': If we consider the presence of deadlines, the situation is not that simple. If deadlines are part of the problem, while the restrictions '1 terminal', '1 truck', and 'no release times', are present, then it is an open question whether this problem is in NPC or not. However, if deadlines

---

1 If there are no deadlines, the release times do not have impact on the complexity, since we can wait with the execution of all orders until all the release times have passed without influencing the quality of the solution.

2 Without the presence of timing constraints, the availability of multiple trucks does not have any impact on the complexity, because any multiple-truck schedule can be transformed into a single truck schedule as follows: first execute the moves of truck 1, then of truck 2, etc.. We are assuming that all trucks return to the terminal at the end of their schedule. However, even if this is not the case, we suspect that the problem remains in $P$. 

---
Figure 3.1: Hasse diagram for complexity of PEHP variants
occur together with multiple trucks or release times, then membership of NPC is certain.

The whole situation would be simplified if the open problem was in NPC; then the restrictions '1 terminal' and 'no deadlines' would determine completely which variants are in NPC, while the other two restrictions would have no influence on the complexity at all. However, from a practical point of view, it would be nice if the open problem was in P, since then an efficient algorithm would exist. Therefore further research on the open problem is interesting and relevant.

In the remainder of this report, we will present complexity results on individual PEHP variants. The variants under consideration are all on the P/NPC borderline in Figure 3.1. In Chapter 4, we consider the simplest problem (the SPEHP), in which all restrictions are present. Consequently, we consider the variant with deadlines and release times (the SPEHPD) in Chapter 5, deadlines and multiple trucks (STPEHPD) in Chapter 6, and finally the variant with multiple terminals (SMTPGEHP) in Chapter 7.
Chapter 4

Planning with one terminal and no time windows

To obtain complexity results for our PEHP variants, we need problems of which the complexity is known. We will however not fully describe these problems but suffice with a short description and a reference. Unfortunately, the notation used in the literature we refer to sometimes interferes with the notation we have introduced. For example, where we use \( d \) to denote a delivery location, in [3] it is used to denote a deadline. To avoid confusion, any symbol used in the literature we refer to which interferes with our notation, gets an extra accent. For example, we will write \( d \) to denote deadlines in the problems of [3].

In this section, we pay attention the simplest planning problem considered in this report.

**Definition 1** A simple pre- and end-haulage problem (SPEHP) is a PEHP such that:

- there is only one terminal: \( |V_{\text{term}}| = \{v_{\text{term}}\} \);
- the time and distance functions are symmetrical: \( \tau_{xy} = \tau_{yx} \) and \( \delta_{xy} = \delta_{yx} \), for all \( x, y \in V \);
- the triangle inequality holds for the time and distance functions: \( \tau_{xy} + \tau_{yz} \geq \tau_{xz} \) and \( \delta_{xy} + \delta_{yz} \geq \delta_{xz} \) for all \( x, y, z \in V \);
- there is only one truck: \( |A| = \{a\} \), \( \text{start}(a) = v_{\text{term}} \); and
- for all \( o_i \in O \) holds that there are no time restrictions \( (1 \leq i \leq n) \):
  - \( T_{s_i} = [\infty, \infty] \);
  - \( w_{s_i} = 0 \);
  - \( T_{d_i} = [\infty, \infty] \); and
  - \( w_{d_i} = 0 \).
We shall show that SPEHP is in P, by reducing it to the minimization version of the weighted bipartite matching problem (WBMP, see [5], page 248). Such a problem consists of two disjoint sets of nodes, \( V' \) and \( U' \), and a cost function \( c_{ij} \) on pairs of nodes. In the WBMP, each node of in \( V' \) should be matched to a node in \( U' \) such that all nodes in \( U' \) are being matched exactly once and the total matching costs are minimized.

The reduction is as follows. Consider the case where \(|O_e| \geq |O_p|\). The case for which \(|O_e| < |O_p|\) can be done analogously. Let \( V' = \{ v_i \mid o_{ei} \in O_e \} \), and \( U' = \{ u_i \mid o_{pi} \in O_p \} \). Let \( c_{ij} \) describe the costs of matching \( v_i \) to \( u_j \):

\[
c_{ij} = \begin{cases} 
\delta_{s_{ei}, s_{pi}}, & \text{if } j \leq |O_p|; \\
\delta_{d_{ei}, s_{pi}}, & \text{if } j > |O_p| 
\end{cases}
\]

The idea is as follows: we match end-haulages to pre-haulages. This should be done as much as possible since it is always cheaper to drive from customer to customer directly instead of via the terminal, because of the triangle inequality. When the number of end-haulages is unequal to the number of pre-haulages, a number of dummy nodes is added to the matching problem (in the description of \( c_{ij} \) given above these are the nodes \( u_{o_{p,i+1}}, \ldots, u_{o_{p,i}} \)). The cost of matching to such a dummy is equal to the cost of driving back to the terminal from the customer. Thus, the matching cost is equal to the ’empty’ distance in the SPEHP.

We show that the WBMP which is the result of the transformation has a solution \( \leq K \) if and only if the SPEHP has a solution

\[
\leq K + \sum_{o_{ei} \in O_e} \delta_{s_{ei}, d_{ei}} + \sum_{o_{pi} \in O_p} \delta_{s_{pi}, d_{pi}}
\]

\( K \) corresponds to the ’empty distance’.

We can transform a solution of the matching problem back to a SPEHP solution. It is sufficient to present such a solution as a series of small schedules, each set consisting of two or three consecutive truck moves, such that each schedule starts and ends in the terminal node. The order in which these small schedules are carried out is not relevant since we do not have any time constraints. The matching of \( u_i \) to \( v_j \) will be written as \( \phi(i) = j \). If \( \phi(i) \leq |O_p| \), then we obtain a truck schedule describing the moves \( v_{t \text{erm}} = s_{ei} \to d_{ci} \to s_{p\phi(i)} \to d_{p\phi(i)} = v_{t \text{erm}} \). The costs of this schedule are

\[
\delta(\sigma_i) = \delta_{s_{ei}, d_{ei}} + \delta_{d_{ei}, s_{p\phi(i)}} + \delta_{s_{p\phi(i)}, d_{p\phi(i)}} = \delta_{s_{ei}, d_{ei}} + \delta_{s_{p\phi(i)}, d_{p\phi(i)}} + c_{\phi(i)}
\]

If \( \phi(i) > |O_p| \), then we obtain a truck schedule describing the moves \( v_{t \text{erm}} = s_{ei} \to d_{ci} \to v_{t \text{erm}} \). The costs of this schedules are

\[
\delta(\sigma_i) = \delta_{s_{ei}, d_{ei}} + \delta_{d_{ei}, s_{ei}} = \delta_{s_{ei}, d_{ei}} + c_{\phi(i)}
\]

There are \( |O_p| \) schedules of the first type and \(|O_e| - |O_p|\) schedules of the second
type. Thus the total costs are

\[
\sum_{i=1}^{\left| \mathcal{O}_o \right|} \delta^{(\sigma_i)} = \sum_{\varphi^{(i)}=1}^{\left| \mathcal{O}_p \right|} \left( \delta_{s_{x_i} d_{x_i}} + \delta_{s_{p_{\varphi^{(i)}}} d_{p_{\varphi^{(i)}}}} + c_{i \varphi^{(i)}} \right) + \sum_{\varphi^{(i)}=\left| \mathcal{O}_p \right|+1}^{\left| \mathcal{O}_o \right|} \left( \delta_{s_{x_i} d_{x_i}} + c_{i \varphi^{(i)}} \right) = \\
\sum_{i=1}^{\left| \mathcal{O}_o \right|} c_{i \varphi^{(i)}} + \sum_{i=1}^{\left| \mathcal{O}_o \right|} \delta_{s_{x_i} d_{x_i}} + \sum_{i=1}^{\left| \mathcal{O}_p \right|} \delta_{s_{p_{\varphi^{(i)}}} d_{p_{\varphi^{(i)}}}}
\]

The first term is equal to the solution to the matching problem and thus equal to \( K \). Thus we proved that the SPEHP has a solution of \( K + Z \) (\( Z \geq 0 \)) if and only if the corresponding WBMP has a solution of \( K \), so SPEHP is in \( \mathbb{P} \).
Chapter 5

Planning with one terminal, release times, and deadlines

In this section, we consider pre- and end-haulage problems with double time constraints: the execution of an order can only start after the release time and it must be finished before the deadline. However, we will restrict ourselves to problems with one terminal and one truck. The simplified pre- and end-haulage problem is as follows.

**Definition 2** A Simple Pre- and End-Haulage Problem with Release times and Deadlines (SPEHPRD) is a PEHP such that:

- there is only one truck: $A = \{a\}$;
- there is only one terminal: $V_{term} = \{v_{term}\}$;
- for all $o_i \in O$ holds:
  - $T_{s_i} = [t_{min,i}, t_{max,i}]$;
  - $w_{s_i} = 0$;
  - $T_{d_i} = [t_{min,i}, t_{max,i}]$; and
  - $w_{d_i} = 0$.

We show that this problem is NP-complete by reduction from the 'sequencing problem with release times and deadlines on one processor' (SPRD) (see [3]).

An SPRD has a set $T'$ of tasks, where each task $t'$ has a length $l(t')$, a release time $r(t')$, and a deadline $d(t')$. The problem is to find a one-processor schedule $\sigma'$ such that $r(t') \leq \sigma'(t')$ and $\sigma'(t') + l(t') \leq d(t')$. The fact that $\sigma'$ is a schedule means that $\sigma'(t') > \sigma(t'')$ implies that $\sigma'(t') \geq \sigma'(t'') + l(t'')$.

The transformation to an SPEHPRD is as follows. The tasks $t'$ are numbered from 1 to $|T'|$. For each task $i$, a node $v_{ci}$ exists in $V_c$; $\delta_{v_{term},v_{ci}} = \tau_{v_{term},v_{ci}} = \delta_{v_{ci},v_{term}} = \tau_{v_{ci},v_{term}} = l(i)$. Furthermore, each task $i$ yields an order $o_i$, where $s_i = v_{term}$, $d_i = v_{ci}$, $t_{min,i} = 2r(i)$, and $t_{max,i} = 2d'(i) - l(i)$. 

13
Now, a feasible schedule $\sigma'$ for the SPRD can be transformed to a schedule for the SPEHPRD by doubling the scheduled times, and adding truck moves from $v_0$ to the terminal after the execution of each order. Thus, a task $i$ which starts at time $\sigma'(i) \geq r(i)$ and finishes at time $\sigma'(i) + l(i) \leq d(i)$, is transformed into an order $i$ starting at time $2\sigma'(i) \geq 2r(i) = t_{\text{min},i}$ and finishing at time $2\sigma'(i) + l(i) \leq 2\sigma'(i) + 2r(i) - l(i) \leq 2d(i) - l(i) = t_{\text{max},i}$. Furthermore, the truck returns to the terminal in time, being at $2\sigma'(i) + 2l(i) \leq 2\sigma'(j)$, where $j$ is the order scheduled next, since $\sigma'(i) + l(i) \leq \sigma'(j)$.

A solution for the SPEHPRD which is the result of the transformation consists of a truck schedule $\sigma(n)$, in which all odd truck moves serve an order. Let order $i$ be served by truck move $f(i)$. Then, the pickup of order $i$ takes place at time $\bar{t}_{a_{f(i)}}$ while the delivery happens at time $\bar{t}^{d}_{a_{f(i)}}$. Let the corresponding schedule for the SPRD be defined as $\sigma'(i) = \frac{1}{2} \bar{t}^{d}_{a_{f(i)}}$. We now show that $\sigma'$ is indeed a solution for the SPRD, given that $\sigma$ is a solution for the SPEHPRD.

1) $\sigma'$ respects the release times:

$$\sigma'(i) = \frac{1}{2} \bar{t}^{d}_{a_{f(i)}} \geq \frac{1}{2} t_{\text{min},i} = \frac{1}{2} 2r(i) = r(i)$$

2) $\sigma'$ respects the deadlines:

$$\sigma'(i) + l(i) = \frac{1}{2} \bar{t}^{d}_{a_{f(i)}} + l(i) = \frac{1}{2} \bar{t}^{d}_{a_{f(i)}} + \tau_{v_{\text{term},v_{ci}}} \leq$$

$$\frac{1}{2} \bar{t}^{d}_{a_{f(i)}} + \frac{1}{2} \tau_{v_{\text{term},v_{ci}}} \leq \frac{1}{2} t_{\text{max},i} + \frac{1}{2} \tau_{v_{\text{term},v_{ci}}} \leq$$

$$d(i) - \frac{1}{2} \bar{t}^{d}_{a_{f(i)}} + \frac{1}{2} \tau_{v_{\text{term},v_{ci}}} = d(i)$$

3) The tasks are scheduled consecutively, not simultaneously, by $\sigma'$: suppose $\sigma'(i) > \sigma'(j)$. Then $\bar{t}^{d}_{a_{f(i)}} > \bar{t}^{d}_{a_{f(j)}}$. Since the moves are numbered consecutively and orders are only served during odd truck moves, we have $f(i) \geq f(j) + 2$, so

$$\bar{t}^{d}_{a_{f(i)}} \geq \bar{t}^{d}_{a_{f(f(i)-1)}} \geq \bar{t}^{d}_{a_{f(f(j)+1)}} \geq$$

$$\bar{t}^{d}_{a_{f(f(j)+1)}} + \tau_{v_{ci}v_{\text{term}}} \geq \bar{t}^{d}_{a_{f(f(j)+1)}} + l(j) \geq$$

$$\bar{t}^{d}_{a_{f(f(j)+1)}} + \tau_{v_{\text{term},v_{ci}}} + l(j) = \bar{t}^{d}_{a_{f(f(j)+1)}} + 2l(j)$$

This implies that

$$\sigma'(i) = \frac{1}{2} \bar{t}^{d}_{a_{f(i)}} \geq \frac{1}{2} \bar{t}^{d}_{a_{f(f(j)+1)}} + l(j) = \sigma'(j) + l(j)$$
5.1 Symmetry and triangle inequality

We remark that SPEHPRDs as constructed above are already symmetric. Furthermore, the roads between customers could only be used for a direct move from or to the terminal, but this can only be the case if the triangle inequality is not satisfied. This would actually ruin the transformation. Thus, although implicit, the SPEHPRD constructed above already satisfies the triangle inequality.

5.2 More time constraints

Although two time constraints have been considered in this section (being release time and deadline), the pickup and the delivery window are equal \( (T_{s_a} = T_{d_a}) \) and the pickup and delivery durations are zero \( (w_{s_a} = w_{d_a} = 0) \). We state that the presence of pickup and delivery durations does not influence the complexity of the PEHP, since they only affect the execution time of an order, so in a reduction they could be modeled as a part of the execution time. Furthermore, we suspect that variants in which other timing constraints are present, for example only a pickup or only a delivery time window, are also NP-complete, as long as there are two time parameters present. The reductions for these problems can probably be done using the SPRD, too.
Chapter 6

Planning with one terminal, deadlines, and multiple trucks

In this section we discuss the complexity of PEHPs with only deadlines. However, whether the single-truck variant of this problem is NP-complete is an open question. Therefore, we will consider multiple trucks. We remark that, although the problem described below considers *any* number of trucks, the problem is also in NPC for a *constant* number of trucks, as long as the constant number is at least 2. This can be seen from the problem we have used for the reduction (the multiprocessor scheduling problem), which is in NPC for any constant number of processors \( \geq 2 \) (see [3]).

This variant of the PEHP is defined as follows.

**Definition 3** A Single Terminal Pre- and End-Haulage Problem with Deadlines (STPEHPD) is a PEHP such that:

- there is only one terminal: \( V_{term} = \{ v_{term} \} \);
- for all \( o_i \in O \) holds:
  - \( T_{s_i} = [-\infty, t_{max,i}] \);
  - \( w_{s_i} = 0 \);
  - \( T_{d_i} = [-\infty, t_{max,i}] \); and
  - \( w_{d_i} = 0 \).

We show that this problem is NP-complete by reduction from the ‘multiprocessor scheduling problem’ (MSP) (see [3]).

An MSP has a set \( T' \) of tasks, where each task \( t' \) has a length of \( l(t') \). There are \( m' \) processors on which these tasks have to be executed before a deadline \( D \). The problem is to find \( m \) schedules such that all tasks are scheduled exactly once and are finished before \( D \).

The transformation to the STPEHPD is as follows. The tasks \( t' \) are numbered from 1 to \( T' \). For each task \( i \), a node \( v_{ci} \) exists in \( V_c \), \( \delta_{v_{term}v_{ci}} = \tau_{v_{term}v_{ci}} = \).
\[ \delta_{v_i; v_{term}} = \tau_{v_{term}; v_{term}} = l(i). \] Furthermore, each task \( i \) yields an order \( o_i \), where \( s_i = v_{term}, d_i = v_{ci} \), and \( t_{max,i} = 2D - l(i) \). Finally, we have \( m' \) trucks, all starting at \( v_{term} \).

The transformation resembles the one in Chapter 5. However, since we have a multiprocessor schedule, we have to transform each processor schedule into a truck schedule; a task scheduled at time \( \sigma'(i) \) on processor \( j \) yields a truck move at time \( 2\sigma'(i) \) from \( v_{term} \) to \( v_{ci} \) (serving order \( o_i \)) and a move back to the terminal. Since the original MSP ends before \( D \), all moves will have ended before \( 2D \), and each order \( i \) executed before \( 2D - l(i) \).

### 6.1 Symmetry and triangle inequality

Since the graph constructed in this section is similar to the one constructed in Section 5, the same arguments with respect to symmetry and triangle inequality apply. Therefore, these two graph properties do not influence the complexity of the STPEHPD.
Chapter 7

Planning with multiple terminals

In this section, we consider pre- and end-haulage problems with multiple terminals but without time constraints. The simplified pre- and end-haulage problem is as follows.

**Definition 4** A simple multiple-terminal pre- and end-haulage problem (SMT-PEHP) is a PEHP such that:

- there is only one truck: \( A = \{ a \} \);
- for all \( o_i \in O \) holds:
  - \( T_{s_i} = [\infty, \infty] \);
  - \( w_{s_i} = 0 \);
  - \( T_{d_i} = [\infty, \infty] \); and
  - \( w_{d_i} = 0 \).

We show that this problem is NP-complete by reduction from the TSP (see [5]). In the TSP, we have a complete graph \( G' = (V', E') \) and \( c_{ij} \) as costs of traveling from \( i \) to \( j \) for all \( i, j \in V' \). The question is how to travel along all nodes such that the total costs are minimized.

The reduction is globally as follows: we introduce a terminal and a customer for each TSP node, plus an order from the terminal to the customer. The costs of the TSP-problem correspond to the distances between an arbitrary terminal and an arbitrary customer, except when they stem from the same TSP node, in which case we choose 1. Formally, we get \( V_c = \{ v_{ci} \mid i \in V' \} \), and
\[ V \text{term} = \{ v_{\text{term},i} \mid i \in V^\prime \}; \]

\[
\delta_{xy} = \begin{cases} 
1, & \text{if } x = v_{\text{term},i} \text{ and } y = v_{s_i}, \text{ for some } i \in V^\prime; \\
1, & \text{if } x = v_{ci} \text{ and } y = v_{\text{term},i}, \text{ for some } i \in V^\prime; \\
c_{ij}, & \text{if } x = v_{\text{term},i}, y = v_{v_{ci}}, \text{ and } i \neq j; \\
c_{ij}, & \text{if } x = v_{ci}, y = v_{\text{term},j}, \text{ and } i \neq j; \\
c_{ij}, & \text{if } x = v_{ci}, y = v_{v_{ci}}, \text{ and } i \neq j; \text{ and} \\
c_{ij}, & \text{if } x = v_{ci}, y = v_{v_{cj}}, \text{ and } i \neq j. 
\end{cases}
\]

Note that the distance function is symmetric and satisfies the triangle inequality.

For the orders, we get \( O = \{ o_i \mid 1 \leq i \leq |V^\prime| \} \), such that \( s_i = v_{\text{term},i} \) and \( d_i = v_{s_i} \), for all \( 1 \leq i \leq |V^\prime| \).

It is not hard to see that a TSP path having costs \( K \) corresponds to a unique truck schedule having distance \( K + |V^\prime| \).

### 7.1 Symmetry and triangle inequality

It is evident that the constructed SMPEHP is symmetric. We will show below that if the original TSP satisfies the triangle inequality, so does the constructed SMPEHP. Since TSP's with triangle inequality are still NP-complete, so is the SMPEHP with triangle inequality.

We will now show that if the TSP has the triangle inequality, the constructed SMPEHP satisfies the triangle inequality, \( \delta_{xy} + \delta_{yz} = \delta_{xz} \) for all \( x, y, z \in V \). We only present the essential cases; the others can be derived using symmetry and the fact that \( \delta_{xx} = 0 \) for all \( x \in V \). The cases are presented by means of the table below. The notation \( v_i \) is used to denote a node in the set \( \{ v_{ci}, v_{\text{term},i} \} \).

The reader can check by inspection that the triangle inequality is satisfied.

<table>
<thead>
<tr>
<th></th>
<th>y</th>
<th>z</th>
<th>( \delta_{xy} )</th>
<th>( \delta_{yz} )</th>
<th>( \delta_{xz} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_{ci} )</td>
<td>( v_{\text{term},i} )</td>
<td>( v_j )</td>
<td>1</td>
<td>( c_{ij} )</td>
<td>( c_{ij} )</td>
</tr>
<tr>
<td>( v_{\text{term},i} )</td>
<td>( v_{ci} )</td>
<td>( v_j )</td>
<td>1</td>
<td>( c_{ij} )</td>
<td>( c_{ij} )</td>
</tr>
<tr>
<td>( v_{ci} )</td>
<td>( v_j )</td>
<td>( v_{\text{term},i} )</td>
<td>( c_{ij} )</td>
<td>( c_{ij} )</td>
<td>1</td>
</tr>
<tr>
<td>( v_{ci} )</td>
<td>( v_{cj} )</td>
<td>( v_{\text{term},j} )</td>
<td>( c_{ij} )</td>
<td>1</td>
<td>( c_{ij} )</td>
</tr>
<tr>
<td>( v_{ci} )</td>
<td>( v_{\text{term},j} )</td>
<td>( v_{cj} )</td>
<td>( c_{ij} )</td>
<td>1</td>
<td>( c_{ij} )</td>
</tr>
<tr>
<td>( v_i )</td>
<td>( v_j )</td>
<td>( v_k )</td>
<td>( c_{ij} )</td>
<td>( c_{jk} )</td>
<td>( c_{ik} )</td>
</tr>
</tbody>
</table>
Chapter 8

Conclusion and further research

We have presented a complexity analysis for the most important PEHP variants. Only the simplest variant appears to be in P. If multiple terminals are allowed, or release times and deadlines, the problem is NP-complete. When there are only deadlines but multiple trucks, the problem is also NP-complete. Whether the latter variant but with only one truck is tractable or not, is still an open question.

As far as complexity is concerned, we only considered membership of P versus NPC. It is however to be expected that some NP-complete variants can be solved in pseudopolynomial time. It would also be interesting to have insight in the existence of bounded approximation algorithms.

Furthermore, there are other variants to be considered. On the one hand, there are variants resembling those often considered for the scheduling problems we used to prove NP-completeness, like constant task length, precedence constraints, and weighted tardiness (see [3]). On the other hand, we could think of variants stemming from the pre- and end-haulage practice, like a periodical order scheme or including separate trailer movements.

The goal of this report however was to show that the development of heuristic algorithms for PEHP’s is relevant because of the complexity of the problem. Since most main variants are NP-complete, an extensive complexity analysis of a large set of PEHP-variants is outside the scope of this report. We conclude by stating that PEHP’s are interesting problems which need to be studied to find good algorithms capable of solving them.
Appendix A

Time windows

A time window $T$ is an interval $[t_1, t_2]$. The empty time window is denoted by $[]$; it is used to denote the impossible timing. The set of all possible time windows $T^+$ is defined as $T^+ = \{[t_1, t_2] \in \mathbb{R}^2 \mid t_2 \geq t_1\}$. The set of all time windows $T$ is defined as $T = T^+ \cup \{[]\}$. The following operations are defined on time windows:

contains:

\[ t \in [t_1, t_2], \]
if and only if $t_1 \leq t \leq t_2$, and $t \not\in []$ for all $t \in \mathbb{R}$;

before/after:

\[ t < [t_1, t_2] \text{ if } t < t_1, \]
\[ t > [t_1, t_2] \text{ if } t > t_2, \]
\[ t \leq [t_1, t_2] \text{ if } t \leq t_2, \]
\[ t \geq [t_1, t_2] \text{ if } t \geq t_1, \]
\[ t \not\in [], t \not\in [], t \not\in [], \text{ and } t \not\in [] \text{ for all } t \in \mathbb{R}, \]
\[ [t_{11}, t_{12}] < [t_{21}, t_{22}] \text{ if } t_{12} < t_{21}, \]
\[ [t_{11}, t_{12}] > [t_{21}, t_{22}] \text{ if } t_{11} > t_{22}, \]
\[ [t_{11}, t_{12}] \leq [t_{21}, t_{22}] \text{ if } t_{12} \leq t_{22}, \]
\[ [t_{11}, t_{12}] \geq [t_{21}, t_{22}] \text{ if } t_{11} \geq t_{21}, \text{ and} \]
\[ T \not\subset [], \text{ and } T \not\supset [], \text{ for all } T \in T, [\cdot] \leq [\cdot], [\cdot] \geq [\cdot], \text{ and} \]
\[ T \not\subset [\cdot] \text{ and } T \not\supset [\cdot] \text{ for all } T \in T^+. \]
addition/subtraction:

\[ [t_1, t_2] \boxplus x = [t_1 + x, t_2 + x], \]
\[ ] \boxplus x = [], \]
\[ [t_1, t_2] \boxdot (x_1, x_2) = [t_1 + x_1, t_2 + x_2], \]
\[ ] \boxdot (x_1, x_2) = [], \]
\[ [t_1, t_2] \boxminus x = [t_1 - x, t_2 - x], \]
\[ ] \boxminus x = [], \]
\[ [t_1, t_2] \boxslash (x_1, x_2) = [t_1 - x_1, t_2 - x_2], \]
\[ ] \boxslash (x_1, x_2) = [] \]

(note that \((x_1, x_2)\) does not have to be a time window since \(x_1\) may be larger than \(x_2\));

intersection:

\[ [t_{11}, t_{12}] \cap [t_{21}, t_{22}] = \begin{cases} [\max(t_{11}, t_{21}), \min(t_{12}, t_{22})], & \text{if } \max(t_{11}, t_{21}) \geq \min(t_{12}, t_{22}), \\ [ ], & \text{otherwise,} \end{cases} \]
\[ ] \cap T = []; \]

union:

\[ [t_{11}, t_{12}] \cup [t_{21}, t_{22}] = \begin{cases} [\min(t_{11}, t_{21}), \max(t_{12}, t_{22})], & \text{if } \max(t_{11}, t_{21}) \geq \min(t_{12}, t_{22}), \\ [ ], & \text{otherwise,} \end{cases} \]
\[ ] \cup T = T; \]

sub:

\[ [t_{11}, t_{12}] \sqsubseteq [t_{21}, t_{22}] \iff t_{11} \geq t_{21} \land t_{12} \leq t_{22}; \]

super:

\[ [t_{11}, t_{12}] \sqsupseteq [t_{21}, t_{22}] \iff t_{11} \leq t_{21} \land t_{12} \geq t_{22}; \]

size:

\[ [t_1, t_2] = t_2 - t_1, [t_1, t_2] \text{ is undefined}; \]

where \(x, x_1, x_2, t_1, t_2, t_{11}, t_{12}, t_{21}, t_{22} \in \mathbb{R}\) and \(T \in \mathcal{T}\).
Bibliography


