Approximation Algorithms for a Vehicle Routing Problem

Sven O. Krumke* Sleman Saliba* Tjark Vredeveld†
Stephan Westphal*

May 2, 2006

Abstract

In this paper we investigate a real-world large-scale vehicle dispatching problem with strict real-time requirements, posed by our cooperation partner, the German Automobile Association (ADAC). Service units starting at their current positions are to serve at most a given number of requests without returning to their homepositions.

We show that the problem of finding a feasible dispatch for service units starting at their current position and serving at most k requests is NP-complete even for $k = 2$.

We also present a polynomial time $(2k - 1)$-approximation algorithm, where again k denotes the maximal number of requests served by a single servie unit. If k equals the total number of requests, we provide a $\left(2 - \frac{1}{k}\right)$-approximation which works similar to the Double-Tree-Algorithm for the metric TSP. Finally, we extend the approximation algorithm to include linear and quadratic lateness costs, which are of interest with respect to the application at ADAC.

1 Introduction

Currently, the German Automobile Association (ADAC) evaluates an automated dispatching system for service vehicles (units) and service contractors (contractors) on the basis of exact cost-reoptimization. This means that a current dispatch is maintained, which contains all known yet unserved requests and which is near optimal on the basis of the current data; whenever a unit becomes idle its next request is read from the current dispatch; at each event (new request, finished service, etc.) the dispatch is updated by a reoptimization run.

A feasible current dispatch for all known requests and available service vehicles is a partition of the requests into tours for units and contractors such that each request is in exactly one tour and each unit drives exactly one tour so that the cost function is minimized. Cost contributions come from driving costs for units, fixed service costs per requests for contractors, and a strictly convex lateness cost for the violation of soft time windows at each request (currently quadratic). The latter cost structure is chosen so as to avoid large individual waiting times for customers. For details we refer to [KJR02, HKR05, KRT02].

The problem we consider in this paper is of the following form. Given a snapshot of currently available service units (vehicles) and a set of requests that have to be served, we are requested to assign up to k requests to a vehicle such that the overall

*University of Kaiserslautern, Department of Mathematics, P.O.Box 3049, Paul-Ehrlich-Str. 14, 67653 Kaiserslautern, Germany. {krumke,saliba,westphal}@mathematik.uni-kl.de

†Maastricht University, Department of Quantitative Economics, P.O.Box 616, 6200 MD, Maastricht, the Netherlands. t.vredeveld@ke.unimaas.nl

costs are minimized. In order for the problem to be feasible, we assume that the total number $|E|$ of requests is at most k times the number $|U|$ of units.

This problem is related to *metric multi-depot vehicle routing problems (MDVRP)* and to *metric k-customer vehicle routing problems*. In a multi-depot vehicle routing problem a fleet of vehicles located at more than one depot are to serve locally dispersed customers such that the vehicles return to one of the depots and the transportation costs are minimized. The difference to our problem is that we do not want our service units to return to their home positions, since by the time the assigned requests are served, new requests have arrived that are to be assigned to service units in the next iteration. The multi-depot vehicle routing problem has been shown to be NP-hard for more than one depot [BCG87].

In the metric *k-customer vehicle routing problem(k-VRP)* all vehicles are based at one depot and are required to serve at most k customers each such that the transportation costs are minimized. It is known that the metric 2-customer vehicle routing problem is polynomially solvable, since it can be transformed to a minimum matching problem, whereas for $k \geq 3$ the metric k-VRP is NP-hard, which was shown by Haimovich and Rinnooy Kan [HRK85].

For our problem, we know that serving at most one customer is easy. Again the problem can be transformed to a matching problem. In contrast, Krumke and Dischke [Dis04] showed that it is hard for $k \geq 3$. The case $k = 2$ was still open. In Section 3 we will prove the NP-completeness of finding a feasible dispatch for vehicles all located at different sites serving at most two customers.

We develop an algorithm that runs in $\mathrm{O}\left(n^3\right)$ time and gives a $(2k-1)$ approximation for the metric problem. Extending this approach to linear and quadratic penalty costs for violating a request's deadline leads to a $\left(\frac{2k^2 r_1}{r_1+1} + 2k - 1\right)$ and a $\left(\frac{2k^3 r_2}{r_2+1} + \frac{2k^2 r_1}{r_1+1} + 2k - 1\right)$ approximation respectively, where r_1 and r_2 are constant factors associated with the linear and quadratic penalty terms, respectively.

In the next section, the exact setting of the problem and the notation used in this paper are introduced. In Section 3 we prove the NP-completeness of the underlying decision problem. The following Section 4 is dedicated to the description of the approximation algorithm and finally we show in Section 5 that after the introduction of lateness costs the presented algorithm is still a constant factor approximation.

2 Problem Definition and Notation

We are given a set of vehicles (or units) U that correspond to our service units, a set of requests E and a metric distance function $d : (U \cup E) \times (U \cup E) \rightarrow \mathbb{R}^+$. A tour consists of a unit $u \in U$ and a sequence of requests $\left(e_{u,1}, e_{u,2}, \ldots, e_{u,h(u)}\right)$, which are visited by vehicle u in the given order. We will denote such a tour by a vector $\left(u, e_{u,1}, e_{u,2}, \ldots, e_{u,h(u)}\right)$.

In the case of the basic problem without lateness costs, the cost $c(e_i, e_j)$ of driving from e_i to e_j corresponds to the metric distance $d(e_i, e_j)$ between the two points. In the generalized case with lateness costs, the cost function is composed of the distance between two points plus a lateness term, which denotes the cost of arriving at a certain time at a request,

$$c\left(e_i, e_j\right) = d\left(e_{u,i}, e_{u,j}\right) + r\left(t_{e_j}\right).$$

In this case, c is not metric anymore.

Let us now define now the problems we will be dealing with in this paper.

Definition 2.1 (Vehicle Dispatching Problem, VDP-k)
Given requests E, units U, costs c as above and a number $k \in \mathbb{N}$ such that $|E| \leq$

$k|U|$, *the vehicle dispatching problem* VDP-k *consists of finding a tour* $\big(u, e_{u,1}, e_{u,2}, \ldots, e_{u,h(u)}\big)$ *for each unit* $u \in U$ *which serves* $h(u) \leq k$ *requests, such that each request is served in exactly one tour and such that the total cost of the tours is minimized.*

3 Complexity

Proof that the decision problem is NP-complete. We will now prove that the decision version of the problem VDP-2, is NP-complete. We use a reduction from the problem 3DM which is known to be NP-complete [GJ79]:

Definition 3.1 (3-Dimensional Matching, 3DM)
Given a set $M \subseteq X \times Y \times Z$*, where* X, Y *and* Z *are disjoint sets having the same number of elements, i. e.,* $|X| = |Y| = |Z| = q$*, does* M *contain a* matching*, that is a subset* $M' \subseteq M$ *such that* $|M'| = q$ *and no two elements of* M' *agree in any coordinate.*

Theorem 3.2 *The decision version of* VDP-k *is* NP*-complete even for* $k = 2$.

Proof:
Given an instance of 3DM, we construct an instance for the decision version of VDP-2 such that 3DM contains a matching M' if and only if there exists a feasible dispatch for VDP-2 with cost at most B.

Let the sets X, Y, and Z with $|X| = |Y| = |Z| = q$ and $M \subseteq X \times Y \times Z$ denote an arbitrary instance of 3DM. The metric space of the instance for VDP-2 is induced by the graph $G = (V, A)$, which is build by connecting the following subgraphs.

For every triple $m_i = (x_i, y_i, z_i) \in M$ we construct a subgraph $G_i = (V_i, A_i)$ with twelve vertices and eighteen edges as shown in Figure 1. A single vertex is defined for every element of the basic sets X, Y, Z that is not contained in any triple.

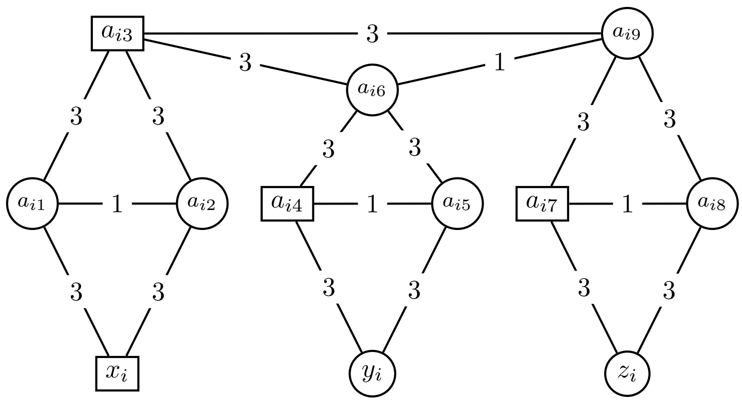

Figure 1: Subgraph $G_i = (V_i, A_i)$
Squared vertices indicate elements contained in U (available units)
Circular vertices indicate elements contained in E (released requests)

The subgraphs and the single vertices respectively are connected by the edges

$$A_c = \{[x_j, x_{j+1}], [y_j, y_{j+1}], [z_j, z_{j+1}], j = 1, \ldots, q-1\}.$$

Note that there is exactly one vertex in $G = (V, A)$ for every element of the basic sets X, Y, and Z and that these vertices are connected by the edge set A_c. Thus the graph $G = (V, A)$ is connected and defined by the vertex set

$$V = X \cup Y \cup Z \cup \bigcup_{i=1}^{m} \{a_{ih} \,:\, h = 1, \ldots, 9\}$$

and the edge set

$$A = A_c \cup \bigcup_{i=1}^{m} A_i.$$

Notice that $|V| = 3q + 9m$ and $|A| = 18m + 3(q-1)$.

We now have to construct the set U corresponding to the position of the available vehicle units and the set E corresponding to the location of the released requests. Hereto, we set

$$U = X \cup \bigcup_{i=1}^{m} \{a_{i3}, a_{i4}, a_{i7}\}$$

and

$$E = Y \cup Z \cup \bigcup_{i=1}^{m} \{a_{i1}, a_{i2}, a_{i5}, a_{i6}, a_{i8}, a_{i9}\}.$$

We assume that all units start their shifts at time $t_0 = 0$ and we do not take any service time or lateness costs into account. The distance between two points, i.e., between two requests and unit's current position, respectively, is defined as the shortest path on the graph G, where the travel time to traverse an edge $e = [v_j, v_k] \in A$ is defined as follows

$$d(v_j, v_k) = \begin{cases} 1 & \text{if } [v_j, v_k] \in \bigcup_{i=1}^{m} \{[a_{i1}, a_{i2}], [a_{i4}, a_{i5}], [a_{i6}, a_{i9}], [a_{i7}, a_{i8}]\}, \\ 3 & \text{otherwise,} \end{cases}$$

and the driving costs are 1 per traveled time unit.

We claim that 3DM contains a matching M' if and only if there exists a feasible dispatch for VDP-2 on the graph G with cost less or equal to $B = 4(q + 3m)$.

By construction, we see that $2|U| = |E|$. Therefore every unit is required to serve exactly two requests in a feasible solution. As each edge of length 1 is incident only to edges of length 3, each possible tour serving two requests costs at least 4 units. Hence, to meet the cost constraint, the cost of each tour cannot exceed 4, since the number of vehicles is $|U| = q + 3m$.

As all edges incident to vertices in Y or Z have length 3, any tour for a unit a_{i4} or a_{i7} serving a request outside of subgraph G_i costs at least 6. Moreover, any tour for a unit $x_i \in X$ serving two requests from different subgraphs has also cost at least 6. Hence, to stay within the budget, each tour can only serve requests within one subgraph.

Therefore, to meet the cost constraint, the only two possible tour combinations within a subgraph G_i are of the following types:

Type 1: (a_{i3}, a_{i6}, a_{i9}), (x_i, a_{i2}, a_{i1}), (a_{i4}, a_{i5}, y_i), (a_{i7}, a_{i8}, z_i),
Type 2: (a_{i3}, a_{i2}, a_{i1}), (a_{i4}, a_{i5}, a_{i6}), (a_{i7}, a_{i8}, a_{i9}).

The order in which requests a_{i2} and a_{i1} are served within a tour can be changed without altering the type of the tour combination.

To prove our claim, first consider a feasible dispatch D for the instance of VDP-2. By the above considerations, we know that each vehicle $u \in U$ operates exactly one tour serving exactly two requests and that the requests in each subgraph, G_i, are served by tour combinations either of type 1 or of type 2. A matching M' for

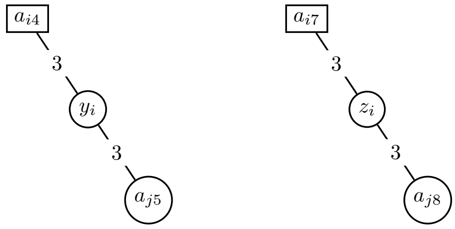

Figure 2: Possible tours between two subgraphs

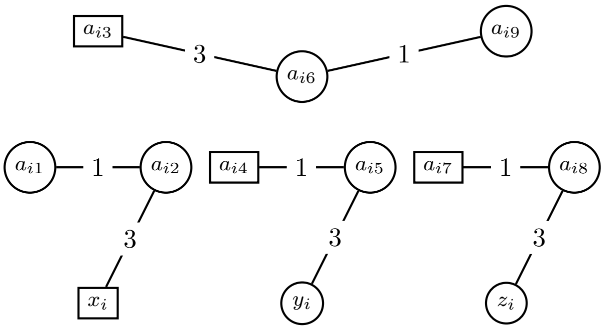

Figure 3: Tours of type 1

the instance of 3DM is obtained by the triples $m_i = (x_i, y_i, z_i)$ for which the nodes $\{a_{i3}, a_{i6}, a_{i9}\}$ form a tour in the dispatch D, i.e., the tours serving the requests in subgraph G_i are of type 1. By definition of a dispatch each element in the basic sets X, Y, and Z occur in at most one tour and therefore in at most one triple m_i. Furthermore, every element of the basic sets are contained in at least one tour. Assume, to the contrary, that there exists an element $w \in X \cup Y \cup Z$ that is not part of a triple $m \in M'$, then all subgraphs containing w are of type 2, which means that w is not part of any tour, contradicting the feasibility of a dispatch.

Conversely, let $M' \subseteq M$ with $|M'| = q$ such that no two triples in M' agree in any coordinate, that is, M' is a feasible matching for the instance of 3DM. The corresponding dispatch is given by choosing tours of type 1 in subgraph G_i whenever $m_i \in M'$ and choosing tours of type 2 in subgraph G_i whenever $m_i \notin M'$. We claim that this dispatch serves each request at a total cost of $4(q + 3m)$. First, we show that each request is served. Within a subgraph G_i every request $V_i \setminus (U \cup \{y_i, z_i\})$ is served by as well tours of type 1 as tours of type 2. For each request $y \in Y$ there is one subgraph G_i with $y = y_i$ that is traversed by tour combination of type 1, as otherwise M' cannot be a matching. Hence, each request $y \in Y$ is served. In the same way, we can show that each request $z \in Z$ is served. Moreover, none of the requests is served more than once. Again, for requests in $V \setminus (U \cup Y \cup Z)$ this holds by construction. Suppose that $w \in Y \cup Z$ is served by two vehicles. As requests in $Y \cup Z$ can only be served by tours of type 1, these tours were generated by two triples of the subset M' both containing w, but this is a contradiction to the definition of the matching M'. In the same way we can show that every vehicle serves at least and at most one tour. It remains to show that the total cost of the tours is bounded by $B = 4(q + 3m)$. By construction, we have q subgraphs corresponding to the q

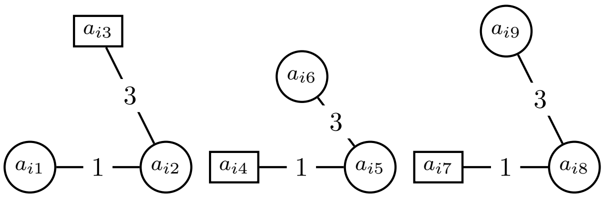

Figure 4: Tours of type 2

triples of the matching M', therefore having tour combinations of type 1 with four tours each of length 4. Moreover, we have $(m - q)$ subgraphs corresponding to the remaining triples of M that are not contained in M'. These subgraphs are traversed by tour combination of type 2 which contain three tours of length 4. Hence, the total cost is $4q \cdot 4 + 3(m - q) \cdot 4 = 4(3m + q)$.

□

4 A constanct factor approximation algorithm

In this section we present a $(2k-1)$-approximation to approximately solve the metric VDP-k in $O(n^3)$ time. If k is constant, this gives a constant factor approximation. Moreover, for $k = n = |E|$ we provide an $2 - \frac{1}{n}$-approximation which works similar to the Double-Tree Algorithm for metric TSPs.

4.1 $k < |E|$

We define the auxiliary graph $G = (V, A)$ with $V = U \cup E \cup \{s, t\}$ and $A = A_1 \cup A_2 \cup A_3$, with $A_1 := \{s\} \times U$ $A_2 := U \times E$ $A_3 := E \times \{t\}$. Define capacities $u : A \to \mathbb{R}$ with $u(a) := k$ for all $a \in A_1$ and $u(a) := 1$ for all $a \in A_2 \cup A_3$. Define costs $c : A \to \mathbb{R}$ with $c((u, e)) := d(u, e)$ for all $a \in A_2$ and $c(a) = 0$ for all $a \in A_1 \cup A_3$. We can find an integral maximal $s - t-$flow of minimum cost in $O(n^3)$ time. This flow corresponds to an assignment of units to events such that every request is assigned to at most one unit and that each unit is assigned to at most k requests. For all $u \in U$ let $E_u := \{e_{u,1}, e_{u,2}, \cdots, e_{u,h(u)}\} \subseteq E$ be the set of requests assigned to u ordered such that $d\,(u, e_{u,i}) \le d(u, e_{u,i+1}) i = 0, \cdots, h - 1$. Our heuristic assigns these requests in the given order to u.

Algorithm 1 Match-Dispatch

Input: A set of units U, a set of requests E, a metric weight function $c : (U \cup E) \times (U \cup E) \to \mathbb{R}^+$.
Output: a set of tours $\mathcal{T}$
$\mathcal{T} := \emptyset$
Construct auxiliary graph G
Compute maximal $s - t-$flow of minimum cost
for all $u \in U$ **do**
    Let $E_u := \{e_{u,1}, e_{u,2}, \cdots, e_{u,h(u)}\}$ be the set of requests assigned to u
    $\mathcal{T} := \mathcal{T} \cup (e_{u,1}, e_{u,2}, \cdots, e_{u,h(u)})$
end for
Return $\mathcal{T}$

Let $M := \{(u, e) | \forall u \in U, e \in E_u\} \subseteq A_2$ be the set of arcs corresponding to the assignment chosen by MATCH-DISPATCH, $D \subseteq A$ be the set of arcs corresponding to the solution obtained thereby and $O \subseteq A$ be the set of arcs chosen by the optimal solution. Then, we can state the following propositions:

Lemma 4.1

$$c(D) \leq (2 - \frac{1}{k}) \cdot c(M)$$

Proof: For all $u \in U$ let $E_u = (e_{u,1}, e_{u,2}, \cdots, e_{u,h(u)})$ be the events covered by u in the given order. Recall that $h(u) \leq k$.

$$\begin{aligned}
c(D) &= \sum_{u \in U} \left(d(u, e_{u,1}) + \sum_{i=2}^{h(u)} d(e_{u,i-1}, e_{u,i}) \right) \\
&\leq \sum_{u \in U} \left(d(u, e_{u,1}) + \sum_{i=2}^{h(u)} \Big(d(u, e_{u,i-1}) + d(u, e_{u,i}) \Big) \right) \\
&= \sum_{u \in U} \left(d(u, e_{u,1}) + \sum_{i=2}^{h(u)} d(u, e_{u,i-1}) + \sum_{i=2}^{h(u)} d(u, e_{u,i}) \right) \\
&= \sum_{u \in U} \left(2 \cdot (\sum_{i=1}^{h(u)-1} d(u, e_{u,i})) + d\left(u, e_{u,h(u)}\right) \right) \\
&= 2 \cdot c_{OPT}(M) - \sum_{u \in U} \left(d\left(u, e_{u,h(u)}\right) \right) \\
&\leq (2 - \frac{1}{k}) \cdot c(M)
\end{aligned}$$

□

Lemma 4.2

$$k \cdot c(O) \geq c(M)$$

Proof: For all $u \in U$ let $E_u = (e_{u,1}, e_{u,2}, \cdots, e_{u,h(u)})$ be the events covered by u in the given order. Recall that $h(u) \leq k$.

$$\begin{aligned}
k \cdot c(O) &= k \cdot \sum_{u \in U} d(u, e_{u,1}) + \sum_{i=2}^{h(u)} d(e_{u,i-1}, e_{u,i}) \\
&\geq \cdot \sum_{u \in U} \sum_{j=1}^{h(u)} \left(d(u, e_{u,1}) + \sum_{i=2}^{j} d(e_{u,i-1}, e_{u,i}) \right) \\
&\geq \cdot \sum_{u \in U} \sum_{j=1}^{h(u)} d(u, e_{u,j}) \\
&\geq c(M)
\end{aligned}$$

□

Theorem 4.3 MATCH-DISPATCH *is a* $2k-1-$ *approximation of the metric VDP-k.*

Proof:

$$\frac{c(D)}{c(O)} \leq \frac{(2 - \frac{1}{k}) \cdot c(M)}{\frac{1}{k} \cdot c(M)} = 2k - 1$$

□

Lemma 4.4 *For $k = 2$ the Approximation ratio of Theorem 4.3 is tight.*

Proof: Consider the example shown in Figure 5. The heuristic assigns the events e_1, e_2 to u_1 and e_3, e_4 to u_2 causing costs of $2 \cdot 3 = 6$, whereas an optimal solution would assign the events e_1, e_3 to u_1 and e_2, e_4 to u_2 causing costs of $2\cdot(1+\epsilon) = 2+2\cdot\epsilon$. For $\epsilon \to 0$ the approximation rate converges to 3. $\square$

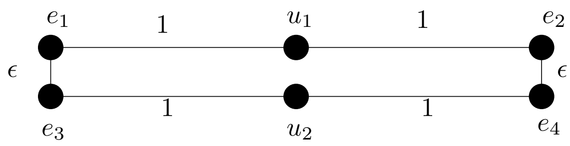

Figure 5: The graph for the proof of Theorem 4.4

Observe, that this example works for the special case of Euclidean distances, too. Lemma 4.4 can be generalized for arbitrary k and therefore it is not possible to analyze it more tightly.

Lemma 4.5 *The Approximation ratio of Theorem 4.3 is tight for arbitrary k.*

Proof: Let $U := \{u_i | i = 1, \cdots, k\}$ be the set of units and $E := \{u_{i,j} | i, j = 1, \cdots, k\}$ the set of requests. Let c be the metric closure induced by the weight function c', with $c'(u_i, e_{i,j}) = 1$ and $c'(e_{j,i}, e_{j+1,i}) = \epsilon$ for all $j = 1, \cdots k-1$, $i = 1, \cdots k$ (see Figure 6). Analogously to the proof of Lemma 4.4 Match-Dispatch chooses to assign $e_{i,j}$ to u_i for all $i, j = 1, \cdots k$ incurring costs of $k \cdot (2 \cdot k - 1)$ whereas it is possible to get hand on a better solution by assigning $e_{i,j}$ to u_j for all $i, j = 1, \cdots k$ incurring costs of $k \cdot (1 + (k-1) \cdot \epsilon)$. Thus, Match-Dispatch is not better than $(2k-1)$-approximative on arbitrary metric systems. $\square$

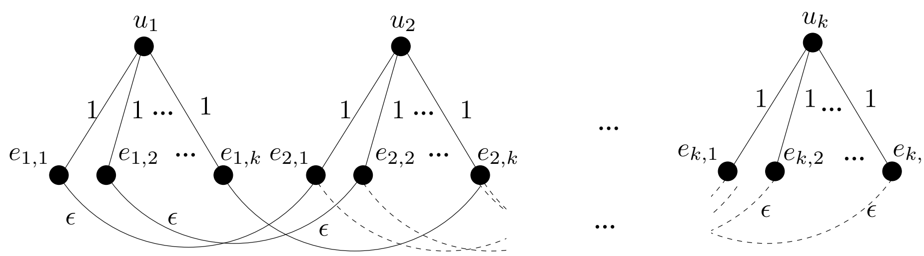

Figure 6: The graph for the proof of Lemma 4.5

4.2 $k = |E|$

In this case, we can provide another algorithm derived from the double-tree approximation for the metric TSP, which is much better than the one of the previous section.

Therefore, we construct an undirected simple graph $G = (V := U \cup E, R := V \times V)$ with a cost function $c : A \to \mathbb{R}^+$ with $c(i, j) = 0$ if $i, j \in U$ and $c(i, j) := d(i, j)$ else. After computing a minimum spanning tree T in G, we remove the edges $U \times U$ obtaining $|U|$ connected components T_u with exactly one element $u \in U$ in each of them. These connected components form the tours. For each u let $v_u \in T_u$ be the event with the maximum distance from u within T_u. Let T'_u be the Graph obtained by doubling the edges of T_u. Find an Eulerian tour S in T'_u such that v_u is the last event being served for the first time within this tour. Since all of the other events have already been served u can stop at v_u.

Algorithm 2 Tree-Dispatch

Input: A set of units U, a set of requests E, a metric weight function $c : (U \cup E) \times (U \cup E) \to \mathbb{R}^+$.
Output: a set of tours $\mathcal{T}$
$\mathcal{T} := \emptyset$
$G = (V := U \cup E, R := V \times V)$
for all $u_i, u_j \in U$ **do**
    $c((u_i, u_j)) := 0$
end for
Compute minimum spanning tree T in G w.r.t c
$A := A \setminus U \times U$
for all $u \in U$ **do**
    Let T_u be the connected component of T with $u \in T_u$
    Let v_u be the event with maximum distance from u within T_u
    Find Eulerian Tour S_u in T'_u with v_u being the last element served
    Let P_{v_u} be the simple $u - v_u$-path in T_u
    $\mathcal{T} := \mathcal{T} \cup (S_u \setminus P_{v_u})$
end for
Return $\mathcal{T}$

For this algorithm we need to show, that:

Lemma 4.6 *It is always possible to find an Eulerian tour S_u in T'_u such that v_u is the last event being served for the first time within this tour.*

Proof: Since c is a nonnegative function, we can assume v_u to be a leaf. Let P_{v_u} be the simple $o_u - v_u$-path in T_u. We can find an Eulerian tour S_u by applying a depth-first-search to T_u and whenever there is a decision to be made which vertex to visit next we rather pick one which is not an element of P_{v_u} than one which is an element of P_{v_u}. This way, we will have reached all other nodes in T_u prior to v_u. □

Lemma 4.7 *For all $u \in U$ it holds that*

$$c(P_{v_u}) \geq \frac{1}{|T_u|} c(T_u)$$

Proof:

$$c(T_u) \leq \sum_{v \in T_u} c(P_v) \leq \sum_{v \in T_u} c(P_{v_u}) \leq |T_u| \cdot c(P_{v_u})$$

□

Let $O \subseteq R$ be the set of arcs chosen by the optimal solution.

Lemma 4.8

$$c(T) \leq c(O)$$

Proof: By adding $|U|-1$ edges to O such that all elements of $|U|$ are connected, we can express O as a spanning tree in G. Since T is a minimum spanning tree the proposition follows directly. $\square$

Theorem 4.9 *For $k = |E|$ TREE-DISPATCH is a $2 - \frac{1}{|E|}$ approximation of the metric VDP-k*

Proof: Due to Lemma 4.8 we can conclude:

$$\begin{aligned}
\frac{c(\mathcal{T})}{c(O)} &\leq \frac{\sum_{u\in U} c(S_u) - c(P_{v_u})}{c(T)} \\
&\overset{Lem.4.7}{\leq} \frac{\sum_{u\in U} 2\cdot c(T_u) - \frac{1}{|T_u|}c(T_u)}{c(T)} \\
&\leq \frac{(2-\frac{1}{|E|})\sum_{u\in U} c(T_u)}{c(T)} \\
&= 2 - \frac{1}{|E|}
\end{aligned}$$

$\square$

5 Extension to lateness costs

The overall goal in the problem posed by our cooperation partner is not only to minimize the incurred driving costs but also to maximize service quality, which means that requests should not be arbitrarily delayed. Therefore we introduce a penalty term punishing deferral of requests. These additional costs will be called *lateness costs*.

Consider Algorithm 1 MATCH-DISPATCH. The metric weight function $c : A \to \mathbb{R}^+$ was defined as

$$c(u,e) = \begin{cases} d(u,e) & \forall a \in A_2, \\ 0 & \forall a \in A_1 \cup A_3. \end{cases} \tag{1}$$

We manipulate the weight function in such a way that it additionally contains a lateness term despite the obvious travel cost. Therefore the weight function $c : A \to \mathbb{R}^+$ reads as follows:

$$c(u,e) = \begin{cases} d(u,e) + r(t(e)) & \forall a \in A_2, \\ 0 & \forall a \in A_1 \cup A_3, \end{cases} \tag{2}$$

where $t(e)$ denotes the vehicle's arrival time at request e. The lateness term $r(t(e))$ is a linear function depending on the travel time in case of linear penalties and a quadratic function in case of quadratic penalties.

5.1 Linear Lateness Costs

As mentioned above, the penalty function $r(t(e))$ depends on the time a service vehicle arrives at a request e. Since we assume the travel time equals the travel distance, we have to take the sequence of the requests within a tour into account. We will do so by multiplying the time used for traveling to a request with a constant factor r_1 and add a constant term r_0. If request e is the l-th request served within a tour, the lateness function reads as follows

$$r(t(e)) = r_1\Big(d(u,e_{u,1}) + \sum_{i=2}^{h(u)} d(e_{u,i-1}, e_{u,i})\Big) + r_0. \tag{3}$$

Considering the new weight function (2) with the lateness term (3), we get the following weights for arcs $a \in A_2$ of the auxiliary graph G

$$c\,(u, e) = d\,(u, e_{u,}) + r_1 d\,(u, e_{u,}) + r_0. \tag{4}$$

Since the cost of a tour is defined as

$$\begin{aligned} c\big(u, e_{u,1}, e_{u,2}, \dots, e_{u,h(u)}\big) = d\,(u, e_{u,1}) + \sum_{i=2}^{h(u)} d\,(e_{u,i-1}, e_{u,i}) + \\ + \sum_{i=1}^{h(u)} \Big(r_1\big(d\,(u, e_{u,1}) + \sum_{j=2}^{i} d\,(e_{u,j-1}, e_{u,j})\,\big) + r_0\Big), \end{aligned} \tag{5}$$

the solution of a dispatch according to Algorithm 1 reads as follows

$$c\,(D) = \sum_{u \in U} \Big(d\,(u, e_{u,1}) + \sum_{i=2}^{h(u)} d\,(e_{u,i-1}, e_{u,i}) + r_1 \sum_{i=1}^{h(u)} \big(d\,(u, e_{u,1}) + \sum_{j=2}^{i} d\,(e_{u,j-1}, e_{u,j})\,\big) + h(u) r_0\Big). \tag{6}$$

Note that $h(u) \leq k$ is the actual number of events in a particular tour.

The *Minimal Cost Flow Problem (MCF)*, which is solved in Algorithm 1 Match-Dispatch to assign at most k requests to a service vehicle $u \in U$, does not take any sequencing of the assigned requests into account. Therefore every single event is seen as the first request served in a tour. Hence, the objective value of the MCF-Problem is

$$c\,(M) = \sum_{u \in U} \Big(\sum_{i=1}^{h(u)} \big(d\,(u, e_{u,i}) + r_1 d\,(u, e_{u,i}) + r_0\big)\Big). \tag{7}$$

Prior to proving the approximation ratio we will state an observation that will be useful in the following computations.

Observation 5.1

$$\sum_{i=1}^{h(u)} \Big(d\,(u, e_{u,1}) + \sum_{j=2}^{i} d\,(e_{u,j-1}, e_{u,j})\,\Big) \leq 2 \sum_{i=1}^{h(u)-1} (h(u) - i)\, d\,(u, e_{u,i}) + \sum_{i=1}^{h(u)} d\,(u, e_{u,i})$$

Proof:

$$\begin{aligned} \sum_{i=1}^{h(u)} \Big(d\,(u, e_{u,1}) + \sum_{j=2}^{i} d\,(e_{u,j-1}, e_{u,j})\,\Big) &= \sum_{i=1}^{h(u)-1} (h(u) - i)\, d\,(e_{u,i}, e_{u,i+1}) + h(u) d\,(u, e_{u,1}) \\ &\leq \sum_{i=1}^{h(u)-1} (h(u) - i)\,\big(d\,(u, e_{u,i}) + d\,(u, e_{u,i+1})\,\big) + h(u) d\,(u, e_{u,1}) \\ &= \sum_{i=1}^{h(u)-1} (h(u) - i)\, d\,(u, e_{u,i}) + \sum_{i=1}^{h(u)-1} (h(u) - i)\, d\,(u, e_{u,i+1}) + h(u) d\,(u, e_{u,1}) \\ &= \sum_{i=1}^{h(u)-1} (h(u) - i)\, d\,(u, e_{u,i}) + \sum_{i=2}^{h(u)} (h(u) - i + 1)\, d\,(u, e_{u,i}) + h(u) d\,(u, e_{u,1}) \\ &= \sum_{i=1}^{h(u)-1} (2h(u) - 2i + 1)\, d\,(u, e_{u,i}) + d\,\big(u, e_{u,h(u)}\big) \\ &= 2 \sum_{i=1}^{h(u)-1} (h(u) - i)\, d\,(u, e_{u,i}) + \sum_{i=1}^{h(u)} d\,(u, e_{u,i}) \end{aligned}$$

□

Lemma 5.2

$$c\left(D\right) \leq \left(2-\frac{1}{k}+\frac{2kr_1}{r_1+1}\right)\cdot c\left(M\right)$$

Proof: For all $u \in U$ let $E_u = (e_1, \cdots, e_{h(u)})$ be the events covered by u in the given order.

$$\begin{aligned}
c\left(D\right) &= \sum_{u\in U}\left(d\left(u,e_{u,1}\right)+\sum_{i=2}^{h(u)} d\left(e_{u,i-1},e_{u,i}\right)+r_1\sum_{i=1}^{h(u)}\Big(d\left(u,e_{u,1}\right)+\sum_{j=2}^{i} d\left(e_{u,j-1},e_{u,j}\right)\Big)+hr_0\right)\\
&\leq \sum_{u\in U}\left(2\cdot\sum_{i=1}^{h(u)-1} d\left(u,e_{u,i}\right)+d\left(u,e_{u,h(u)}\right)+r_1\sum_{i=1}^{h(u)}\Big(d\left(u,e_{u,1}\right)+\sum_{j=2}^{i} d\left(e_{u,j-1},e_{u,j}\right)\Big)+h(u)r_0\right)\\
&\overset{\text{Obs.(5.1)}}{\leq} \sum_{u\in U}\left(2\cdot\sum_{i=i}^{h(u)-1} d\left(u,e_{u,i}\right)+d\left(u,e_{u,h(u)}\right)+2r_1\sum_{i=1}^{h(u)-1}\left(h(u)-i\right)d\left(u,e_{u,i}\right)+r_1\sum_{i=1}^{h(u)} d\left(u,e_{u,i}\right)+h(u)r_0\right)\\
&= 2c\left(M\right)+\sum_{u\in U}\left(-d\left(u,e_{u,h(u)}\right)-r_1\sum_{i=1}^{h(u)} d\left(u,e_{u,i}\right)-h(u)r_0+2r_1\sum_{i=1}^{h(u)-1}\left(h(u)-i\right)d\left(u,e_{u,i}\right)\right)\\
&\leq \left(2-\frac{1}{k}\right)c\left(M\right)+\sum_{u\in U}\left(-r_1\sum_{i=1}^{h(u)-1} d\left(u,e_{u,i}\right)-\left(h(u)-1\right)r_0+2r_1\sum_{i=1}^{h(u)-1}\left(h(u)-i\right)d\left(u,e_{u,i}\right)\right)\\
&= \left(2-\frac{1}{k}+2k\right)c\left(M\right)+\sum_{u\in U}\Bigg(-r_1\sum_{i=1}^{h(u)-1} d\left(u,e_{u,i}\right)-\left(h(u)-1\right)r_0-2k\sum_{i=1}^{h(u)} d\left(u,e_{u,i}\right)-\\
&\qquad 2kr_1 d\left(u,e_{u,h(u)}\right)-2kh(u)r_0-2r_1\sum_{i=1}^{h(u)-1} id\left(u,e_{u,i}\right)\Bigg)
\end{aligned}$$

Since

$$\sum_{u\in U}2k\sum_{i=1}^{h(u)} d\left(u,e_{u,i}\right)=\frac{2k}{r_1+1}c\left(M\right)-\sum_{u\in U}\Big(\frac{2k}{r_1+1}h(u)r_0\Big),$$

we have

$$\begin{aligned}
c\left(D\right) &\leq \left(2-\frac{1}{k}+2k-\frac{2k}{r_1+1}\right)c\left(M\right)+\sum_{u\in U}\Bigg(-r_1\sum_{i=1}^{h(u)-1} d\left(u,e_{u,i}\right)-\left(h(u)-1\right)r_0-\\
&\qquad 2kr_1 d\left(u,e_{u,h(u)}\right)+\frac{2k}{r_1+1}h(u)r_0-2kh(u)r_0-2r_1\sum_{i=1}^{h(u)-1} id\left(u,e_{u,i}\right)\Bigg)
\end{aligned}$$

and

$$c\left(D\right) \leq \left(2-\frac{1}{k}+\frac{2kr_1}{r_1+1}\right)c\left(M\right),$$

because

$$-r_1\sum_{i=1}^{h(u)-1} d\left(u,e_{u,i}\right)-\left(h(u)-1\right)r_0-2kr_1 d\left(u,e_{u,h(u)}\right)+\frac{2k}{r_1+1}h(u)r_0-2kh(u)r_0-2r_1\sum_{i=1}^{h(u)-1} id\left(u,e_{u,i}\right)\leq 0.$$

□

Lemma 5.3

$$kc\left(O\right) \geq c\left(M\right)$$

Proof: For all $u \in U$ let $E_u = (e_1, \cdots, e_{h(u)})$ be the events covered by u in the given order.

$$
\begin{aligned}
k \cdot c(O) = k \sum_{u \in U} \Bigg(& \sum_{i=1}^{h(u)} d(e_{u,i-1}, e_{u,i}) + d(u, e_{u,1}) + \\
& \sum_{i=1}^{h(u)} r_1 \Big(\sum_{i=2}^{i} d(e_{u,j-1}, e_{u,j}) + d(u, e_{u,1}) \Big) + h(u) r_0 \Bigg) \\
\geq \sum_{u \in U} \Bigg(& \sum_{j=1}^{h(u)} \Big(\sum_{i=2}^{j} d(e_{u,i-1}, e_{u,i}) + d(u, e_{u,1}) + \\
& \sum_{i=1}^{h(u)} r_1 \sum_{j=1}^{i} \Big(\sum_{l=2}^{j} d(e_{u,l-1}, e_{u,l}) + d(u, e_{u,1}) \Big) + h(u) r_0 \Bigg) \\
\geq \sum_{u \in U} \Bigg(& \sum_{i=1}^{(u)} d(u, e_{u,i}) + r_1 \sum_{i=1}^{(u)} d(u, e_{u,i}) + h(u) r_0 \Bigg) \\
= c(M) &
\end{aligned}
$$

□

Theorem 5.4 MATCH-DISPATCH *with linear lateness cost is a* $\left(\frac{2k^2 r_1}{r_1+1} + 2k - 1\right)$*−approximation of VDP-*$k_{min}$*.*

Proof:

$$
\frac{c(D)}{c(O)} \leq \frac{\left(\frac{2kr_1}{r_1+1} + 2 - \frac{1}{k}\right) \cdot c(M)}{\frac{1}{k} \cdot c(M)} = \frac{2k^2 (r_1)}{r_1 + 1} + 2k - 1
$$

□

5.2 Quadratic Lateness Costs

The case of quadratic lateness penalties is especially important, since our cooperation partner utilizes such functions to ensure quality of service. Hence, extending the linear case to the quadratic is crucial for the success of our work. We model this requirement by squaring the driving time needed to reach a request and multiplying this term with a constant factor $r_2 \geq 0$. Adding the linear and constant lateness terms lead to the following penalty function for request e awaiting service at position l in a tour

$$
r(t(e)) = r_2 \Big(d(u, e_{u,1}) + \sum_{i=2}^{h(u)} d(e_{u,i-1}, e_{u,i}) \Big)^2 + r_1 \Big(d(u, e_{u,1}) + \sum_{i=2}^{h(u)} d(e_{u,i-1}, e_{u,i}) \Big) + r_0. \tag{8}
$$

Therefore we

$$
c(u, e_u) = d(u, e_{u,}) + r_2 d(u, e_{u,}) + r_1 d(u, e_{u,}) + r_0 \tag{9}
$$

get as new weight function for arcs $a \in A_2$ in the auxiliary graph G.

The cost of a tour, the solution of a dispatch and the objective value of the MCF-Problem according to Algorithm 1 Match-Dispatch are composed in the same way as in the linear case by adding the appropriate quadratic terms

$$\begin{aligned}c\left(u, e_{u,1}, \ldots, e_{u,h(u)}\right) = d\left(u, e_{u,1}\right) + \sum_{i=1}^{h} d\left(e_{u,i-1}, e_{u,i}\right) + \\ + r_2 \sum_{j=1}^{h(u)} \Big(d\left(u, e_{u,1}\right) + \sum_{i=2}^{j} d\left(e_{u,j-1}, e_{u,j}\right)\Big)^2 \\ + r_1 \sum_{j=1}^{h(u)} \Big(d\left(u, e_{u,1}\right) + \sum_{i=2}^{j} d\left(e_{u,j-1}, e_{u,j}\right)\Big) + hr_0,\end{aligned} \tag{10}$$

$$\begin{aligned}c\left(D\right) = \sum_{u \in U} \Bigg(d\left(u, e_{u,1}\right) + \sum_{i=2}^{h(u)} diji - 1i + r_2 \sum_{j=1}^{h(u)} \Big(d\left(u, e_{u,1}\right) + \sum_{i=2}^{j} d\left(e_{u,i-1}, e_{u,i}\right)\Big)^2 + \\ + r_1 \sum_{j=1}^{h(u)} \Big(d\left(u, e_{u,1}\right) + \sum_{i=2}^{j} d\left(e_{u,i-1}, e_{u,i}\right)\Big) + h(u) r_0\Bigg)\end{aligned} \tag{11}$$

$$c\left(M\right) = \sum_{u \in U} \left(\sum_{i=1}^{h(u)} \Big(d\left(u, e_{u,i}\right) + r_2 d\left(u, e_{u,i}\right)^2 + r_1 d\left(u, e_{u,i}\right) + r_0\Big)\right) \tag{12}$$

We can now show that Algorithm 1 Match-Dispatch is still a constant factor approximation independent of the number of service vehicle $|U|$ and the number of requests $|E|$. It only depends on the maximal number of requests of a tour and the constant factors r_2, r_1 and r_0.

Lemma 5.5

$$c\left(D\right) \le \left(2 - \frac{1}{k} + \frac{2kr_1}{r_1 + 1} + \frac{2kr_2}{r_2 + 1}\right) \cdot c\left(M\right)$$

Proof: We will assume that $d\left(u, e_{u,i}\right)^2 \ge d\left(u, e_{u,i}\right)$. This is always the case, if $d\left(u, e_{u,1}\right) \ge 1$, otherwise scale the distances appropriately. Moreover, we need some important observations to prove the lemma.

$$\Big(\sum_{i=1}^{j} d\left(u, e_{u,i}\right)\Big)^2 \le j \sum_{i=1}^{j} d\left(u, e_{u,i}\right)^2 \tag{13}$$

and

$$\sum_{j=1}^{h(u)} d\left(u, e_{u,j}\right) \sum_{i=1}^{j-1} d\left(u, e_{u,i}\right) \le \sum_{j=1}^{h(u)} \left(j - 1\right) d\left(u, e_{u,j}\right)^2. \tag{14}$$

Using these equations, we can show for the quadratic term

$$\begin{aligned}
\sum_{j=1}^{h(u)} \Big(d\left(u, e_{u,1}\right) + \sum_{i=2}^{j} d\left(e_{u,i-1}, e_{u,i}\right) \Big)^2 &\leq \sum_{j=1}^{h(u)} \Big(d\left(u, e_{u,1}\right) + \sum_{i=2}^{j} \left(d\left(u, e_{u,i-1}\right) + d\left(u, e_{u,i}\right)\right) \Big)^2 \\
&= \sum_{j=1}^{h(u)} \Big(2 \sum_{i=1}^{j-1} d\left(u, e_{u,i}\right) + d\left(u, e_{u,j}\right) \Big)^2 \\
&= \sum_{j=1}^{h(u)} \bigg(4 \Big(\sum_{i=1}^{j-1} d\left(u, e_{u,i}\right) \Big)^2 + 4 d\left(u, e_{u,j}\right) \sum_{i=1}^{j-1} d\left(u, e_{u,i}\right) + d\left(u, e_{u,j}\right)^2 \bigg) \\
&\overset{by (13)\&(14)}{\leq} 4 \sum_{j=1}^{h(u)} (j-1) \sum_{i=1}^{j-1} d\left(u, e_{u,i}\right)^2 + 4 \sum_{j=1}^{h(u)} (j-1)\, d\left(u, e_{u,j}\right)^2 + \sum_{j=1}^{h(u)} d\left(u, e_{u,j}\right)^2 \\
&= 4 \sum_{i=1}^{h(u)-1} \left(\frac{k\left(k-1\right)}{2} - \frac{i\left(i-1\right)}{2} \right) d\left(u, e_{u,i}\right)^2 + 4 \sum_{i=1}^{h(u)} (i-1)\, d\left(u, e_{u,i}\right)^2 + \sum_{i=1}^{h(u)} d\left(u, e_{u,i}\right)^2 \\
&= 2 \sum_{i=1}^{h(u)-1} \big(k\left(k-1\right) - i\left(i-1\right) \big) d\left(u, e_{u,i}\right)^2 + 4 \sum_{i=1}^{h(u)} (i-1)\, d\left(u, e_{u,i}\right)^2 + \sum_{i=1}^{h(u)} d\left(u, e_{u,i}\right)^2 .
\end{aligned} \tag{15}$$

Therefore we have

$$\begin{aligned}
c\left(D\right) = & \sum_{u\in U}\Bigg(d\left(u,e_{u,1}\right)+\sum_{i=2}^{h(u)}d\left(e_{u,i-1},e_{u,i}\right)+r_1\sum_{j=1}^{h(u)}\Big(d\left(u,e_{u,1}\right)+\sum_{i=2}^{j}d\left(e_{u,j-1},e_{u,j}\right)\Big)+\\
& +r_2\sum_{j=1}^{h(u)}\Big(d\left(u,e_{u,1}\right)+\sum_{i=2}^{j}d\left(e_{u,i-1},e_{u,i}\right)\Big)^2+h(u)r_0\Bigg)\\
\le & \sum_{u\in U}\Bigg(2\sum_{i=1}^{h(u)-1}d\left(u,e_{u,i}\right)+d\left(u,e_{u,h(u)}\right)+2r_1\sum_{i=1}^{h(u)-1}\left(h(u)-i\right)d\left(u,e_{u,i}\right)+r_1\sum_{i=1}^{h(u)}d\left(u,e_{u,i}\right)+\\
& +2r_2\sum_{i=1}^{h(u)-1}\big(k\left(k-1\right)-i\left(i-1\right)\big)d\left(u,e_{u,i}\right)^2+4r_2\sum_{i=1}^{h(u)}\left(i-1\right)d\left(u,e_{u,i}\right)^2+r_2\sum_{i=1}^{h(u)}d\left(u,e_{u,i}\right)^2+h(u)r_0\Bigg)\\
\le & \left(2-\frac{1}{k}\right)c\left(M\right)+\\
& +\sum_{u\in U}\Bigg(-2r_1\sum_{i=1}^{h(u)-1}d\left(u,e_{u,i}\right)-r_1d\left(u,e_{u,h(u)}\right)-2r_2\sum_{i=1}^{h(u)-1}d\left(u,e_{u,i}\right)^2-r_2d\left(u,e_{u,h(u)}\right)^2-\left(h(u)-1\right)r_0+\\
& +2r_1\sum_{i=1}^{h(u)-1}\left(h(u)-i\right)d\left(u,e_{u,i}\right)+r_1\sum_{i=1}^{h(u)}d\left(u,e_{u,i}\right)+\\
& +2r_2\sum_{i=1}^{h(u)-1}\big(k\left(k-1\right)-i\left(i-1\right)\big)d\left(u,e_{u,i}\right)^2+4r_2\sum_{i=1}^{h(u)}\left(i-1\right)d\left(u,e_{u,i}\right)^2+r_2\sum_{i=1}^{h(u)}d\left(u,e_{u,i}\right)^2\Bigg)\\
= & \left(2-\frac{1}{k}+2k+2k^2\right)c\left(M\right)+\\
& +\sum_{u\in U}\Bigg(-2r_1\sum_{i=1}^{h(u)-1}d\left(u,e_{u,i}\right)-r_1d\left(u,e_{u,h(u)}\right)-2r_2\sum_{i=1}^{h(u)-1}d\left(u,e_{u,i}\right)^2-r_2d\left(u,e_{u,h(u)}\right)^2-\left(h(u)-1\right)r_0-\\
& -2k\sum_{i=1}^{h(u)}d\left(u,e_{u,i}\right)-2kr_2\sum_{i=1}^{h(u)}d\left(u,e_{u,i}\right)^2-2kr_1d\left(u,e_{u,h(u)}\right)-2kh(u)r_0\\
& -2k^2\sum_{i=1}^{h(u)}d\left(u,e_{u,i}\right)-2k^2r_1\sum_{i=1}^{h(u)}d\left(u,e_{u,i}\right)-2k^2r_2d\left(u,e_{u,h(u)}\right)^2-2k^2h(u)r_0\\
& +2r_1\sum_{i=1}^{h(u)-1}\left(-i\right)d\left(u,e_{u,i}\right)+r_1\sum_{i=1}^{h(u)}d\left(u,e_{u,i}\right)+\\
& +2r_2\sum_{i=1}^{h(u)-1}\big(-k-i\left(i-1\right)\big)d\left(u,e_{u,i}\right)^2+4r_2\sum_{i=1}^{h(u)}\left(i-1\right)d\left(u,e_{u,i}\right)^2+r_2\sum_{i=1}^{h(u)}d\left(u,e_{u,i}\right)^2\Bigg)
\end{aligned}$$

Since we assumed that $d\left(u,e_{u,i}\right)^2\ge d\left(u,e_{u,i}\right)$, we get

$$-2k^2\sum_{i=1}^{h(u)}d\left(u,e_{u,i}\right)^2\le-2k^2\sum_{i=1}^{h(u)}d\left(u,e_{u,i}\right)=-\frac{2k^2}{r_2+1}c\left(M\right)+\frac{2k^2r_1}{r_2+1}\sum_{i=1}^{h(u)}d\left(u,e_{u,i}\right)+\frac{2k^2}{r_2+1}h(u)r_0 \tag{16}$$

and

$$-2k\sum_{i=1}^{h(u)}d\left(u,e_{u,i}\right)=-\frac{2k}{r_1+1}c\left(M\right)+\frac{2kr_2}{r_1+1}\sum_{i=1}^{h(u)}d\left(u,e_{u,i}\right)^2+\frac{2k}{r_1+1}h(u)r_0. \tag{17}$$

Therefore

$$c\left(D\right) \leq \left(2 - \frac{1}{k} + 2k + 2k^2 - \frac{2k}{r_1+1} - \frac{2k^2}{r_2+1}\right) c\left(M\right) +$$

$$+ \sum_{u \in U} \Bigg(-2r_1 \sum_{i=1}^{h(u)-1} d\left(u, e_{u,i}\right) - r_1 d\left(u, e_{u,h(u)}\right) - 2r_2 \sum_{i=1}^{h(u)-1} d\left(u, e_{u,i}\right)^2 - r_2 d\left(u, e_{u,h(u)}\right)^2 - \left(h(u)-1\right) r_0 -$$

$$- 2kr_2 \sum_{i=1}^{h(u)} d\left(u, e_{u,i}\right)^2 - 2kr_1 d\left(u, e_{u,h(u)}\right) - 2kh(u)r_0 + \frac{2kr_2}{r_1+1} \sum_{i=1}^{h(u)} d\left(u, e_{u,i}\right)^2 + \frac{2k}{r_1+1} h(u)r_0$$

$$- 2k^2 r_1 \sum_{i=1}^{h(u)} d\left(u, e_{u,i}\right) - 2k^2 r_2 d\left(u, e_{u,h(u)}\right)^2 - 2k^2 h(u)r_0 + \frac{2k^2 r_1}{r_2+1} \sum_{i=1}^{h(u)} d\left(u, e_{u,i}\right) + \frac{2k^2}{r_2+1} h(u)r_0$$

$$+ 2r_1 \sum_{i=1}^{h(u)-1} \left(-i\right) d\left(u, e_{u,i}\right) + r_1 \sum_{i=1}^{h(u)} d\left(u, e_{u,i}\right) +$$

$$+ 2r_2 \sum_{i=1}^{h(u)-1} \big(-k - i\left(i-1\right) \big) d\left(u, e_{u,i}\right)^2 + 4r_2 \sum_{i=1}^{h(u)} \left(i-1\right) d\left(u, e_{u,i}\right)^2 + r_2 \sum_{i=1}^{h(u)} d\left(u, e_{u,i}\right)^2 \Bigg)$$

Since

$$2kr_2 \sum_{i=1}^{h(u)} d\left(u, e_{u,i}\right)^2 + 2kh(u)r_0 \geq \frac{2kr_2}{r_1+1} \sum_{i=1}^{h(u)} d\left(u, e_{u,i}\right)^2 + \frac{2k}{r_1+1} h(u)r_0,$$

$$2k^2 r_1 \sum_{i=1}^{h(u)} d\left(u, e_{u,i}\right) + 2k^2 h(u)r_0 \geq \frac{2k^2 r_1}{r_2+1} \sum_{i=1}^{h(u)} d\left(u, e_{u,i}\right) + \frac{2k^2}{r_2+1} h(u)r_0,$$

$$2r_1 \sum_{i=1}^{h(u)-1} i d\left(u, e_{u,i}\right) + r_1 d\left(u, e_{u,h(u)}\right) \geq r_1 \sum_{i=1}^{h(u)} d\left(u, e_{u,i}\right),$$

$$2r_2 \sum_{i=1}^{h(u)-1} k d\left(u, e_{u,i}\right)^2 \geq 4r_2 \sum_{i=1}^{h(u)} \left(i-1\right) d\left(u, e_{u,i}\right)^2,$$

$$2r_2 \sum_{i=1}^{h(u)-1} d\left(u, e_{u,i}\right)^2 + r_2 d\left(u, e_{u,h(u)}\right)^2 \geq r_2 \sum_{i=1}^{h(u)} d\left(u, e_{u,i}\right)^2$$

and all remaining terms are negative, we finally get

$$c\left(D\right) \leq \left(2 - \frac{1}{k} + \frac{2kr_1}{r_1+1} - \frac{2k^2 r_2}{r_2+1}\right) c\left(M\right). \tag{18}$$

□

Lemma 5.6

$$k \cdot c(O) \geq c(M) \tag{19}$$

Proof: For all $u \in U$ let $E_u = (e_{u,1}, \cdots, e_{u,h(u)})$ be the events covered by u in the given order.

$$
\begin{aligned}
k \cdot c\,(O) = k \sum_{u \in U} \Bigg(& \sum_{j=1}^{h(u)} d\,(e_{u,j-1}, e_{u,j}) + d\,(u, e_{u,1}) + \\
& \sum_{j=1}^{h(u)} r_2 \Big(\sum_{i=2}^{j} d\,(e_{u,i-1}, e_{u,i}) + d\,(u, e_{u,1}) \Big)^2 + \\
& \sum_{j=1}^{h(u)} r_1 \Big(\sum_{i=2}^{j} d\,(e_{u,i-1}, e_{u,i}) + d\,(u, e_{u,1}) \Big) + h(u) r_0 \Bigg) \\
\geq \sum_{u \in U} \Bigg(& \sum_{j=1}^{|E_u|} \Big(\sum_{i=2}^{j} d\,(e_{u,i-1}, e_{u,i}) + d\,(u, e_{u,1}) + \\
& \sum_{j=1}^{|E_u|} r_2 \sum_{i=1}^{j} \Big(\sum_{l=2}^{i} d\,(e_{u,l-1}, e_{u,l}) + d\,(u, e_{u,1}) \Big)^2 + \\
& \sum_{j=1}^{|E_u|} r_1 \sum_{i=1}^{j} \Big(\sum_{l=2}^{i} d\,(e_{u,l-1}, e_{u,l}) + d\,(u, e_{u,1}) \Big) + h(u) r_0 \Bigg) \\
\geq \sum_{u \in U} \Bigg(& \sum_{i=1}^{|E_u|} d\,(u, e_{u,i}) + r_2 \sum_{i=1}^{|E_u|} d\,(u, e_{u,i})^2 + r_1 \sum_{i=1}^{|E_u|} d\,(u, e_{u,i}) + |E_u| r_0 \Bigg) \\
= c\,(M) &
\end{aligned}
$$

□

Putting lemmas 5.5 and 5.6 together we can finally prove the constant approximation ratio of Algorithm 1 Match-Dispatch.

Theorem 5.7 Match-Dispatch *with quadratic lateness cost is a* $\big(\frac{2k^3 r_2}{r_2+1} + \frac{2k^2 r_1}{r_1+1} + 2k - 1\big)$*−approximation of VDP-$k_{min}$.*

Proof:

$$
\frac{c(D)}{c(O)} \leq \frac{\big(\frac{2k^2 r_2}{r_2+1} + \frac{2k r_1}{r_1+1} + 2 - \frac{1}{k}\big) \cdot c(M)}{\frac{1}{k} \cdot c(M)} = \frac{2k^3 r_2}{r_2+1} + \frac{2k^2\,(r_1)}{r_1+1} + 2k - 1
$$

□

6 Conclusions and further research

In this paper, we proved NP-completeness for the vehicle dispatching problem Vdp-k even for the case when the length of the tours is restricted to $k = 2$. We also developed an approximation algorithm for the metric version of the problem and extended it to the cases of linear and quadratic lateness functions.

As we have seen in an example, the approximation factor of $(2k - 1)$ in the metric version results from the minimum cost flow problem solved in Algorithm Match-Dispatch. In terms of worst-case analysis, it does not matter whether afterwards we sequence requests optimally, by Match-Dispatch or by a traveling salesman heuristic. However, in practice this sequencing should matter. This is subject to further research and evaluation on real-world problems. An implementation of our algorithms with an extensive evaluation is in progress.

A direction of further research could be an average-case analysis of the problem under a reasonable probabilistic model. As recent results in [COKN05a, COKN05b] about the average-case complexity of dial-a-ride problems indicate, there is hope that a "typical instance" of the problem might be polynomially solvable. Another starting point for further research could be the sequencing subproblem occuring for each vehicle in our algorithm: We are given a vehicle and k customers with soft time windows in which service should be accomplished. Violating the time windows incur additional costs that may be modeled by linear or quadratic functions. Find the tour with minimal operational costs. This problem can be seen as a TSP with soft time windows and lateness penalties.

References

[BCG87] A A Bertossi, P Carraresi, and G Gallo. On some matching problems arising in vehicle scheduling models. *Networks*, 11, 1987.

[COKN05a] A. Coja-Oghlan, S. O. Krumke, and T. Nierhoff. A hard dial-a-ride problem that is easy on average. *Journal on Scheduling*, 8(3):197–210, 2005.

[COKN05b] A. Coja-Oghlan, S. O. Krumke, and T. Nierhoff. A heuristic for the stacker crane problem on trees which is almost surely exact. *Journal of Algorithms*, 2005. A preliminary version appeared in the Proceedings of the 14th International Symposium on Algorithms and Computation, 2004, vol. 2906 of Lecture Notes in Computer Science.

[Dis04] I. Dischke. Disposition von Einsatzfahrzeugen: Startheuristiken, Branching-Regeln und Rundungstechniken. Diplomarbeit, TU Berlin, 2004.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability (A guide to the theory of* NP*-completeness).* W.H. Freeman and Company, New York, 1979.

[HKR05] B. Hiller, S. O. Krumke, and J. Rambau. Reoptimization gaps versus model errors in online-dispatching of service units. *Discrete Applied Mathematics*, 2005. A preliminary version appeared in the Proceedings of the Latin-American Conference on Combinatorics, Graphs and Algorithms, 2004.

[HRK85] M Haimovich and Alexander H G Rinnooy Kan. Bounds and heuristics for capacitated routing problems. *Mathematics of Operations Research*, 10, 1985.

[KJR02] S. O. Krumke and L. M. Torres J. Rambau. Online dispatching of automobile in real-time. ZIB Report 02-18, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2002.

[KRT02] S. O. Krumke, J. Rambau, and L. M. Torres. Real-time dispatching of guided and unguided automobile service units with soft time windows. In *Proceedings of the 10th Annual European Symposium on Algorithms*, volume 2461 of *Lecture Notes in Computer Science*, pages 637–648. Springer, 2002.