Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

Tobias Harks[1] Stefan Heinz
Marc E. Pfetsch Tjark Vredeveld[2]

Online Multicommodity Routing with Time Windows

[1]Institute of Mathematics, Technical University Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany.

[2]Maastricht University, Department of Quantitative Economics,
P.O. Box 616, 6200 MD Maastricht, The Netherlands.

ZIB-Report 07-22 (August 2007)

ONLINE MULTICOMMODITY ROUTING WITH TIME WINDOWS

TOBIAS HARKS, STEFAN HEINZ*, MARC E. PFETSCH,
AND TJARK VREDEVELD

ABSTRACT. We consider a multicommodity routing problem, where demands are released *online* and have to be routed in a network during specified time windows. The objective is to minimize a time and load dependent convex cost function of the aggregate arc flow.

First, we study the fractional routing variant. We present two online algorithms, called SEQ and SEQ^2. Our first main result states that, for cost functions defined by polynomial price functions with nonnegative coefficients and maximum degree d, the competitive ratio of SEQ and SEQ^2 is at most $(d+1)^{d+1}$, which is tight. We also present lower bounds of $(0.265\,(d+1))^{d+1}$ for any online algorithm. In the case of a network with two nodes and parallel arcs, we prove a lower bound of $(2-\frac{1}{2}\sqrt{3})$ on the competitive ratio for SEQ and SEQ^2, even for affine linear price functions. Furthermore, we study resource augmentation, where the online algorithm has to route less demand than the offline adversary.

Second, we consider unsplittable routings. For this setting, we present two online algorithms, called U-SEQ and U-SEQ^2. We prove that for polynomial price functions with nonnegative coefficients and maximum degree d, the competitive ratio of U-SEQ and U-SEQ^2 is bounded by $O(1.77^d\, d^{d+1})$. We present lower bounds of $(0.5307\,(d+1))^{d+1}$ for any online algorithm and $(d+1)^{d+1}$ for our algorithms.

Third, we consider a special case of our framework: online load balancing in the ℓ_p-norm. For the fractional and unsplittable variant of this problem, we show that our online algorithms are p and $O(p)$ competitive, respectively. Such results where previously known only for scheduling jobs on restricted (un)related parallel machines.

1. INTRODUCTION

In this paper, we consider a multicommodity routing problem, in which sets of commodities arrive over time and have to be routed in a network. Each commodity is only alive during a specified time window. We study two variants: (i) the demand of a commodity can be split along several paths; (ii) single path (unsplittable) routing. In all cases, once a demand is routed, no rerouting is allowed. The cost of routing the next small unit of flow on an arc is determined by a nondecreasing and continuous price function of the flow that has already been routed on this arc. For a fixed point in time, the routing cost on an arc is defined by the integral over the arc flow at that time with respect to its price function. Given a certain time span, the total routing cost on an arc is obtained by integrating the routing cost over

---

Date: December 28, 2007.

*Supported by the DFG Research Center MATHEON *Mathematics for key technologies* in Berlin.

time. The goal is to find a routing with minimal cost. Since the demands arrive over time, both problems (splittable and unsplittable) have a natural online variant, which we study in this paper. An online algorithm learns about the existence of a routing request only at its release time, i.e., the lower limit of its time window. We evaluate the quality of online algorithms by *competitive analysis* [24, 21], which has become a standard yardstick for measuring performance. An algorithm ALG is said to be c-competitive, if for any instance the cost of the solution produced by ALG is not more than c times the optimal solution value. The infimum over all such constants c is the *competitive ratio* of ALG.

For both variants of the problem, we define two greedy online algorithms, which we call SEQ and SEQ^2 for the splittable variant and U-SEQ and U-SEQ^2 for the unsplittable variant. Algorithms SEQ and U-SEQ greedily compute a routing decision for *all* commodities released at the same time by minimizing the cost associated with these requests. The other two algorithms SEQ^2 and U-SEQ^2 consider the commodities released at the same time one by one, computing a minimum cost routing for each individual commodity. We investigate cases in which SEQ (U-SEQ) and SEQ^2 (U-SEQ^2) are c-competitive for some constant c.

The problem under investigation arises, for instance, in an inter-domain Quality of Service (QoS) market, in which multiple service providers offer network resources (capacity) to enable Internet traffic with specific QoS constraints, see for example Yahaya and Suda [27] and Yahaya, Harks, and Suda [26]. In such a market, each service provider advertises prices for resources that he wants to sell. Buying providers reserve capacity along paths to route demand (coming from own customers) from source to destination via domains of other providers. The routing of a demand along paths is fixed by establishing a binding contract between the source domain and all domains along the paths. The price for a unit bandwidth changes dynamically with the total bandwidth that is currently in use, that is, routing flow of unit size prompts an update of arc prices. It is assumed that price updates on an arc are determined by a nondecreasing and continuous price function. Harks et al. [17, 18] considered the problem without time windows. In practice, demand requests between service providers come with a lifetime and service providers may have several demand requests at the same time, which allows for a coordinated routing of these demands. In this paper, we explicitly address these issues and generalize the model of Harks et al. [17, 18], see Section 6 for a comparison.

Another application of our framework is the *load balancing problem* in intra-domain networks. In this problem, traffic demands have to be routed from entry nodes to exit nodes of the domain. A natural goal for a domain operator is to distribute the load across the network, i.e., minimize the ℓ_p-norm of the vector of the arc loads in the network, see for instance Fortz and Thorup [14]. Since traffic demands are usually not known beforehand, this problem has a natural online variant, which we study in this paper.

1.1. Related Work

Yahaya et al. [27, 26] empirically studied the performance of a greedy single-path routing protocol for a fixed network topology. This problem was first

Table 1: Lower bounds (LB) and upper bounds (UB) on the competitive ratio of any deterministic online algorithm and on $\textsc{Seq}^2$ for *affine linear price functions* and without time windows as discussed in Harks et al. [18]; here n is the number of commodities.

	splittable		unsplittable	
	any alg.	$\textsc{Seq}$, $\textsc{Seq}^2$	any alg.	U-$\textsc{Seq}$, U-$\textsc{Seq}^2$
LB	$\frac{4}{3}$	$\max\{\frac{2n-1}{n}, 4\}$	2	$3 + 2\sqrt{2}$
UB	–	$\frac{4n^2}{(n+1)^2}$	–	$3 + 2\sqrt{2}$

modeled as an online multicommodity flow problem in Harks et al. [17, 18]. They analyzed a greedy online algorithm (which is closely related to $\textsc{Seq}^2$), and proved that this algorithm is asymptotically 4-competitive for affine price functions; an overview of the results discussed in [18] is given in Table 1. Moreover, they showed some general lower bounds on the competitive ratio of any online algorithm.

In the context of traffic engineering, multicommodity routing problems have been studied by e.g. Fortz and Thorup [13, 14]. There, the goal is to route given demand subject to capacity constraints in order to balance the total load in the network. In this setting, a central planer has full knowledge of all demands, which is not the case in our approach.

Whenever there is no central planer, routing decisions have to be made by selfish users. This has been extensively studied using game theoretic concepts, cf. Roughgarden and Tardos [23], Correa, Schulz, and Stier Moses [10], Altman, Basar, Jimenez, and Shimkin [2], and the references therein. These works study the efficiency of Nash equilibria. The difference to our model is the notion of an equilibrium, which allows for rerouting of demands upon release of a new demand. In our model, once a routing decision has been made, it remains unchanged. In addition, in this paper, the total cost of a multicommodity flow is time dependent, that is, the total cost depends on the time windows of the commodities.

Farzad et al. [12] consider selfish routing problems, where the travel times of agents on an arc are defined by a priority order, see also the thesis of Olver [22]. The total cost on an arc is defined as the integral of the arc flow with respect to given latency functions. They proved tight upper bounds of $(d + 1)^{d+1}$ on the price of anarchy for polynomial latency functions with nonnegative coefficients. For unsplittable flows, they showed a tight upper bound of $3 + 2\sqrt{2}$ for affine latencies and an upper bound of $O(2^d\, d^{d+1})$ for polynomial latency functions. Their model, however, does not capture time varying traffic demands. The results of Farzad et al. [12] are tailored to polynomial latency (price) functions, whereas our results are based on a general technique, which is also applicable to general continuous and non-decreasing price functions. Furthermore, our upper bounds for polynomial price functions in the unsplittable variant improve upon the bounds derived in [12]. The lower bounds presented in Farzad et al. [12] carry over to lower bounds for our algorithms.

Harks and Végh [19] study an online version of the selfish routing problem. Demands are released in an online fashion, and for every released demand a

Table 2: Known lower bounds (LB) and upper bounds (UB) on the competitive ratio of any deterministic online algorithm and on Seq^2 for *polynomial price functions* of degree d.

	splittable		unsplittable	
	any alg.	Seq, Seq^2	any alg.	U-Seq, U-Seq^2
LB	$(0.265\,(d+1))^{d+1}$	$(d+1)^{d+1}$	$(0.5307\,(d+1))^{d+1}$	$(d+1)^{d+1}$
UB	–	$(d+1)^{d+1}$	–	$O(1.77^d\,d^{d+1})$

Nash solution is constructed. Their work is restricted to splittable flows and does not cover time varying demands.

Online load balancing problems on parallel machines can also be seen as a multicommodity routing problem; see Albers [1] for an overview of online algorithms and load balancing in particular. In the simplest form, load balancing can be seen as routing n unsplittable demands from the same source to the same destination over m parallel links. More sophisticated topologies are considered in Section 4.4. Awerbuch et al. [4] considered a greedy online algorithm, where the goal is to minimize the ℓ_p-norm of the aggregated server loads. In particular, they proved an upper bound on the competitive ratio of $1+\sqrt{2}$ for the ℓ_2-norm and $O(p)$ for the ℓ_p-norm. In the same context, Avidor et al. [3], Azar et al. [6], Caragiannis et al. [8], and Suri et al. [25] strengthened the analysis of [4] for special cases (e.g., identical machines, integral jobs, temporary jobs). Their setting, however, is restricted to m parallel arcs and all released jobs have to be assigned to exactly one machine (arc).

In the context of online routing, Awerbuch et al. [5] present online routing algorithms that maximize throughput under the assumption that routings are irrevocable. They present competitive bounds that depend on the number of nodes in the network.

1.2. Our results

The contributions of this paper can be summarized as follows.

First, we study a variant, where demands can be split. For this setting, we investigate the two online algorithms Seq and Seq^2 by means of competitive analysis. We show that every upper bound of Seq carries over to Seq^2, and every lower bound of Seq^2 carries over to Seq. Since Harks et al. [17, 18] showed that for general price functions no online algorithm can be competitive, we consider classes of restricted price functions. Our main result states that for polynomial price functions with nonnegative coefficients and degree d, the competitive ratio of both algorithms is bounded by $(d+1)^{d+1}$. In particular, we prove an upper bound of 4 for general continuous, nondecreasing, and concave price functions. Using a construction presented by Farzad et al. [12], it can be shown that the upper bounds for our algorithms are tight.

We also present lower bounds of $(0.265\,(d+1))^{d+1}$ for *any* online algorithm, matching our upper bound up to a constant factor. If we restrict the input graph to two nodes connected by parallel arcs, we state a lower bound of $(2-\frac{1}{2}\sqrt{3})$ for Seq and for Seq^2, even for affine linear price functions.

Furthermore, we consider a form of resource augmentation, in which SEQ has less demand to route thus weakening the power of the offline adversary. When the price functions are polynomials and the adversary has to route demands which are increased by a factor $\gamma \geq 1$, we present an upper bound on the competitive ratio for SEQ and for SEQ^2 of $\frac{(d+1)^{d+1}}{(d+1)\,\gamma - d}$.

Second, we study unsplittable routings, where demands must be routed on a single path. For this setting, we analyze the competitiveness of the online algorithms U-SEQ and U-SEQ^2. We again show that every upper bound of U-SEQ carries over to U-SEQ^2 and every lower bound of U-SEQ^2 carries over to U-SEQ. Our main result states that for polynomial price functions with nonnegative coefficients and degree d, the competitive ratio of both algorithms is bounded by $O(1.77^d\, d^{d+1})$. In particular, we prove an upper bound of $(1+\sqrt{2})^2$ for affine linear price functions and an upper bound of $4 + 2\sqrt{3}$ for continuous, nondecreasing, and concave price functions.

Based on a reduction from online load balancing problems in the context of scheduling parallel related machines (see Awerbuch et al. [4], Suri, Toth, and Zhou [25] and Caragiannis et al. [8]), we prove a lower bound on the competitive ratio of $(0.5307\,(d+1))^{d+1}$ for *any* online algorithm.

Finally, we study online load balancing problem in general networks. Here, the goal is to route commodities online so as to minimize the ℓ_p-norm of the arc loads. We show that this problem is a special case of our model and investigate the online algorithms SEQ and U-SEQ for the splittable and unsplittable variant in general networks. Our main results in this setting are upper bounds on the competitive ratio of p and $O(p)$ for SEQ and U-SEQ, respectively. To the best of our knowledge, such bounds were previously known only in the context of scheduling problems, see Awerbuch et al. [4], Caragiannis et al. [8], and Suri, Toth, and Zhou [25].

An overview of some lower and upper bounds discussed in this paper is presented in Table 2.

2. Problem Description

We consider the following multicommodity routing problem. Given is a directed network $D = (V, A)$ and nondecreasing and continuous price functions $p_a : \mathbb{R}_+ \to \mathbb{R}_+$ for each arc $a \in A$. Furthermore, a set $\mathcal{K} = \{1, \ldots, n\}$ of commodities is given. Each commodity $j \in \mathcal{K}$ is specified by a time window $[\tilde{\tau}_j, T_j)$, where $\tilde{\tau}_j$ is the release time and T_j is the expiring time, and a bandwidth request (s_j, t_j, d_j). Here $s_j, t_j \in V$ are the source and destination of the request, respectively, and $d_j > 0$ corresponds to the required bandwidth or capacity that needs to be reserved on paths from s_j to t_j in the graph D during the time window $[\tilde{\tau}_j, T_j)$. We assume that $0 \leq \tilde{\tau}_j \leq T_j$ for all $j \in \mathcal{K}$.

Let the *excess flow* in a vertex v for a commodity $j \in \mathcal{K}$ be denoted by

$$\gamma_v^{(j)} = \begin{cases} d_j & \text{if } v = s_j, \\ -d_j & \text{if } v = t_j, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Then, for commodity $j \in \mathcal{K}$, a *feasible flow*, or *routing decision*, is a nonnegative vector $\boldsymbol{f}^{(j)} \in \mathbb{R}^{A}_{+}$, satisfying,

$$\sum_{a \in \delta^+(v)} f_a^{(j)} - \sum_{a \in \delta^-(v)} f_a^{(j)} = \gamma_v^{(j)},$$

where $\delta^+(v)$ and $\delta^-(v)$ are the arcs leaving and entering v, respectively. Note that, we assume that the demand can be split arbitrarily, that is, each infinitesimal amount of the demand can be routed over its own path from s_j to t_j in the network. The routing of the demand for commodity j stays fixed from the release time $\tilde{\tau}_j$ until the expiring time T_j.

When routing an infinitesimal amount of demand on an arc $a \in A$, which has total reserved bandwidth, or load, ℓ_a, this amount of bandwidth incurs a marginal cost of $p_a(\ell_a)$ per time unit for this arc. The total cost on arc a is given by the integral of p_a over the total flow using a. As the bandwidth for a commodity only needs to be reserved during the corresponding time window, the load on an arc varies over time and thus also the costs per arc. Given a feasible flow $\boldsymbol{f} = (\boldsymbol{f}^{(1)}, \ldots, \boldsymbol{f}^{(n)})$, the cost associated at time t with arc a is

$$\int_0^{\sum_{j \in L(t)} f_a^{(j)}} p_a(z)\, dz,$$

where $L(t) = \{j \in \mathcal{K} : \tilde{\tau}_j \le t < T_j\}$ is the set of commodities *alive* at time t. The total cost of the feasible flow is defined as

$$C(\boldsymbol{f}) = \sum_{a \in A} \int_0^{\infty} \int_0^{\sum_{j \in L(t)} f_a^{(j)}} p_a(z)\, dz\, dt.$$

This expression of the total cost can be achieved by single path routing for arbitrarily small demands, see Harks et al. [18] for the case without time windows. We call the problem of finding a feasible flow $\boldsymbol{f}$ that minimizes $C(\boldsymbol{f})$ the Multicommodity Routing problem with Time Windows (MRTW). Note that we do not have any capacity restrictions for the arcs.

2.1. Problem Reformulation

Since an online algorithm has to route all requests with the same release time immediately, one can view the commodities as partitioned into rounds. This viewpoint yields an equivalent representation of the cost function (see Lemma 2.1 below) that we develop in the following and is needed for the description of the online algorithms that we study in this paper.

We define the number of different release times as $R := |\{\tilde{\tau}_1, \ldots, \tilde{\tau}_n\}|$ and write $\{\tau_1, \ldots, \tau_R\} = \{\tilde{\tau}_1, \ldots, \tilde{\tau}_n\}$ with $0 \le \tau_1 < \tau_2 < \cdots < \tau_R$. The set of commodities for each round $r \in \mathcal{R} := \{1, \ldots, R\}$ is denoted by

$$\mathcal{K}_r := \{j \in \mathcal{K} : \tilde{\tau}_j = \tau_r\}$$

with $n_r := |\mathcal{K}_r|$. The maximum expiring date of a round r is denoted by $T_{r,\max} := \max\{T_j : j \in \mathcal{K}_r\}$. Every commodity belongs to exactly one round, that is, $\mathcal{K}_1 \cup \cdots \cup \mathcal{K}_R = \mathcal{K}$, and for two rounds r and r' ($r \neq r'$) we have $\mathcal{K}_r \cap \mathcal{K}_{r'} = \varnothing$.

For a round $r \in \mathcal{R}$ and a time point t, let $J_r(t) = \{j \in \mathcal{K}_r \,:\, \tau_r \le t < T_j\}$ be the set of commodities belonging to round r that are still alive at time t, and let $I_r(t) = \{j \in J_1(t) \cup \dots \cup J_{r-1}(t) \,:\, \tau_r \le t < T_j\}$ be the set of commodities that have been released before round r and are still alive at time t. Note that for $t < \tau_r$ we have $I_r(t) = J_r(t) = \varnothing$. The set $L(t)$ of commodities alive at time t can be rewritten as $L(t) = J_1(t) \cup \dots \cup J_R(t)$. For notational convenience we define $K_r(t) = I_r(t) \cup J_r(t)$ as the set of commodities of rounds $1, \dots, r$ that are still alive at time t.

Consider a round $r \in \mathcal{R}$ and a feasible flow $\boldsymbol{f}$. The total capacity used on arc a at time $t \in [\tau_r, T_{r,\max})$ due to the requests released before round r is denoted by

$$F_{a,r}(\boldsymbol{f}, t) = \sum_{j \in I_r(t)} f_a^{(j)},$$

and by

$$G_{a,r}(\boldsymbol{f}, t) = \sum_{j \in J_r(t)} f_a^{(j)}$$

we denote the amount of flow on arc $a \in A$ at time $t \in [\tau_r, T_{r,\max})$ due to the requests of this round.

To each round $r \in \mathcal{R}$ we associate costs of a feasible routing decision $\boldsymbol{f}$. We denote by $\boldsymbol{f}^r = (\boldsymbol{f}^{(j)} \,:\, j \in \mathcal{K}_r)$ the flow vectors of round r, and we write $\boldsymbol{f}_a^r = (f_a^{(j)} \,:\, j \in \mathcal{K}_r)$ for the vector of all flow values on arc a for round r. The cost $\tilde{C}_a^r(\boldsymbol{f}_a^r, t; \boldsymbol{f}_a^1, \dots, \boldsymbol{f}_a^{r-1})$ associated to round r at time t on arc $a \in A$ is given by

$$\int_0^{\sum_{j \in J_r(t)} f_a^{(j)}} p_a\Big(\sum_{j \in I_r(t)} f_a^{(j)} + z\Big)\, dz \;=\; \int_0^{G_{a,r}(\boldsymbol{f},t)} p_a\big(F_{a,r}(\boldsymbol{f}, t) + z\big)\, dz.$$

The *total cost* associated to round r is obtained by integrating the cost on arc a over the time interval corresponding to r and summing this value for all arcs:

$$C^r(\boldsymbol{f}^r; \boldsymbol{f}^1, \dots, \boldsymbol{f}^{r-1}) := \sum_{a \in A} \int_{\tau_r}^{T_{r,\max}} \tilde{C}_a^r(\boldsymbol{f}_a^r, t; \boldsymbol{f}_a^1, \dots, \boldsymbol{f}_a^{r-1})\, dt.$$

The relation between the cost associated to the rounds $1, \dots, R$ and the total cost of the full routing decision is given in the following lemma.

Lemma 2.1. *Let $\boldsymbol{f}$ be a feasible flow. Then the cost of this feasible flow is equal to the sum of the costs associated to each round, i.e.,*

$$C(\boldsymbol{f}) = \sum_{r=1}^{R} C^r(\boldsymbol{f}^r; \boldsymbol{f}^1, \dots, \boldsymbol{f}^{r-1}).$$

Proof. Consider a feasible routing decision $\boldsymbol{f}$. Then the corresponding cost of this flow is

$$C(\boldsymbol{f}) = \sum_{a\in A}\int_0^\infty \int_0^{\sum_{j\in L(t)} f_a^{(j)}} p_a(z)\,dz\,dt = \sum_{a\in A}\int_0^\infty \int_0^{\sum_{r=1}^{R}\sum_{j\in J_r(t)} f_a^{(j)}} p_a(z)\,dz\,dt$$

$$= \sum_{a\in A}\sum_{r=1}^{R}\int_0^\infty \int_{\sum_{i=1}^{r-1}\sum_{j\in J_i(t)} f_a^{(j)}}^{\sum_{i=1}^{r}\sum_{j\in J_i(t)} f_a^{(j)}} p_a(z)\,dz\,dt.$$

For $t \notin [\tau_r, T_{r,\max})$, we have $J_r(t) = \varnothing$. Hence, the lower and upper bounds of the inner integral are equal, implying that it evaluates to 0. Therefore, for round r, we may restrict the time integral to $[\tau_r, T_{r,\max})$. Furthermore, using that $I_r(t) = J_1(t) \cup \ldots \cup J_{r-1}(t)$, we can write

$$= \sum_{a\in A}\sum_{r=1}^{R}\int_{\tau_r}^{T_{r,\max}} \int_{\sum_{j\in I_r(t)} f_a^{(j)}}^{\sum_{j\in K_r(t)} f_a^{(j)}} p_a(z)\,dz\,dt$$

$$= \sum_{r=1}^{R}\sum_{a\in A}\int_{\tau_r}^{T_{r,\max}} \int_{F_{a,r}(\boldsymbol{f},t)}^{F_{a,r}(\boldsymbol{f},t)+G_{a,r}(\boldsymbol{f},t)} p_a(z)\,dz\,dt$$

$$= \sum_{r=1}^{R} C^r(\boldsymbol{f}^r; \boldsymbol{f}^1, \ldots, \boldsymbol{f}^{r-1}),$$

which completes the proof. □

Observe that we can partition the time horizon into intervals on which the flows are constant: Consider the set of all *events* $E := \{\tilde{\tau}_j, T_j : j \in \mathcal{K}\}$, and let $0 \le \hat{\tau}_1 < \cdots < \hat{\tau}_{|E|}$ be the sorted sequence of all events, i.e., $E = \{\hat{\tau}_1, \ldots, \hat{\tau}_{|E|}\}$. Then, the sets $I_r(t)$, $J_r(t)$, and the functions $F_{a,r}(\boldsymbol{f}, t)$ and $G_{a,r}(\boldsymbol{f}, t)$ are constant on the segments $[\hat{\tau}_i, \hat{\tau}_{i+1})$ $(i = 1, \ldots, |E| - 1)$. Hence,

$$C^r(\boldsymbol{f}^r; \boldsymbol{f}^1, \ldots, \boldsymbol{f}^{r-1}) = \sum_{a\in A}\sum_{i=1}^{|E|-1} (\hat{\tau}_{i+1} - \hat{\tau}_i)\, \tilde{C}_a^r(\boldsymbol{f}_a^r, \hat{\tau}_i; \boldsymbol{f}_a^1, \ldots, \boldsymbol{f}_a^{r-1}). \quad (2)$$

This relation allows to replace the time integral by a sum and will be used several times in this paper.

Remark 2.2. In the following we consider a slightly extended version of the MRTW model in which the rounds are further refined, i.e., we allow for $R \ge |\{\tilde{\tau}_1, \ldots, \tilde{\tau}_n\}|$ rounds with corresponding release times $\tau_1, \ldots, \tau_R$ that are (weakly) sorted as $0 \le \tau_1 \le \tau_2 \le \cdots \le \tau_R$, i.e., rounds with equal release time are now possible. Here, it is understood that the order in which the rounds arrive, is given by their numbering $1, \ldots, R$. Hence, only the online version of MRTW is affected. In fact, the online algorithms discussed in this paper naturally work in this generalized model.

This more general model can be approximated by the original model within arbitrary precision in the following sense. We use a small $\varepsilon > 0$ and for all rounds $1 \leq r, \ldots, r+k \leq R$ with

$$\tau_{r-1} < \tau_r = \cdots = \tau_{r+k} < \tau_{r+k+1}$$

we use modified release times $\tau_r' = \tau_r$, $\tau_{r+1} = \tau_r + \varepsilon, \ldots, \tau_{r+k} = \tau_r + (k-1) \cdot \varepsilon$ (for convenience, we set $\tau_0 = -\infty$ and $\tau_{R+1} = \infty$). The usage of ε guarantees that the rounds are routed sequentially in the order $1, \ldots, R$. Then for any feasible flow $\boldsymbol{f}$ the cost of $\boldsymbol{f}$ in the approximated model tends to the cost of $\boldsymbol{f}$ in the extended model for ε tending to 0 (note that $\boldsymbol{f}$ is feasible for both models).

Remark 2.3. MRTW as just described generalizes the work of Harks et al. [17, 18] in two ways. First, in each round more than one commodity is allowed to be routed. Second, the commodities now come with a given lifetime. More precisely, the setting of [17, 18] is the special case with $n_r = 1$, $\tau_r = 0$ for all rounds $r \in \mathcal{R}$, and $T_j = 1$ for all $j \in \mathcal{K}$. In this case, (2) evaluates to $C^r(\boldsymbol{f}^r) = \sum_{a \in A} \tilde{C}^r(\boldsymbol{f}_a^r, 0)$, and hence the integral over time can be skipped.

3. Online Algorithms

We study two online algorithms, which we call Seq and Seq^2. For commodities in the same round, Seq greedily computes a routing decision, minimizing the cost for these demands. More formally, for each round $r \in \mathcal{R}$ and given the flows $\boldsymbol{f}^1, \ldots, \boldsymbol{f}^{r-1}$, fixed in the previous rounds, Seq solves the following mathematical program

$$\begin{aligned} \min \quad & C^r(\boldsymbol{f}^r; \boldsymbol{f}^1, \ldots, \boldsymbol{f}^{r-1}) \\ \text{s.t.} \quad & \sum_{a \in \delta^+(v)} f_a^{(j)} - \sum_{a \in \delta^-(v)} f_a^{(j)} = \gamma_v^{(j)}, && \forall\, v \in V,\, j \in \mathcal{K}_r \qquad (3) \\ & f_a^{(j)} \geq 0 && \forall\, a \in A,\, j \in \mathcal{K}_r, \end{aligned}$$

to obtain the routing decision $\boldsymbol{f}^r = (\boldsymbol{f}^{(j)} : j \in \mathcal{K}_r)$. Here $\gamma_v^{(j)}$ is defined as in (1).

Note that in this definition of Seq, the rounds do not have to arrive in chronological order, nor need the commodities of one round have to have the same release time. For the online problem, however, we of course need these conditions.

The second algorithm Seq^2 behaves in a similar way as Seq. Whereas Seq routes all commodities in a round at the same time, Seq^2 arbitrarily orders the commodities of a round and routes them sequentially. Formally, Seq^2 constructs a new instance of MRTW with n rounds, each consisting of a single commodity, and then applies Seq on the new instance. The order of the commodities can be obtained as in Remark 2.2. Note that Seq^2 can construct the new rounds in an online manner.

Remark 3.1. Due to the fact that the functions $p_a(\cdot)$ $(a \in A)$ are nonnegative and nondecreasing, the cost functions $C^r(\cdot)$ are convex. As the flow-conserving constraints in (3) are linear constraints, the mathematical program (3) is a convex program and can be solved within arbitrary precision in polynomial time, see e.g., Grötschel et al. [15].

3.1. Characterization of Seq and Seq^2

Since the cost function is convex, we know that any local minimum is also a global minimum and vice versa. To show that a certain flow is optimal for C^r, we need to compute the gradient. Straightforward calculations show that for $r \in \mathcal{R}$, $j \in \mathcal{K}_r$, and $a \in A$:

$$\frac{\partial C^r}{\partial f_a^{(j)}}(\boldsymbol{f}^r; \boldsymbol{f}^1, \dots, \boldsymbol{f}^{r-1}) = \int_{\tau_r}^{T_j} p_a\big(F_{a,r}(\boldsymbol{f}, t) + G_{a,r}(\boldsymbol{f}, t)\big)\, dt.$$

Here and in the following, we use the vector $\boldsymbol{f}$ of all flows on the right hand side, although the expression does not depend on the flows of rounds larger than r.

The necessary and sufficient conditions for the flow of round r to minimize the cost C^r, given the routing decisions of the previous rounds are as in the following lemma.

Lemma 3.2 ([11])**.** *Let $\boldsymbol{f}^1, \dots, \boldsymbol{f}^{r-1}$ be feasible flows for the commodities of the first $r-1$ rounds. Then a feasible flow $\boldsymbol{f}^r = (\boldsymbol{f}^{(j)} : j \in \mathcal{K}_r)$ for round $r \in \mathcal{R}$ solves the convex program* (3) *if and only if for each commodity $j \in \mathcal{K}_r$ and any feasible flow $\boldsymbol{x}^{(j)}$ for this commodity, the following inequality holds*

$$\sum_{a \in A} \Big[\int_{\tau_r}^{T_j} p_a\big(F_{a,r}(\boldsymbol{f}, t) + G_{a,r}(\boldsymbol{f}, t)\big)\, dt\Big]\big(f_a^{(j)} - x_a^{(j)}\big) \leq 0. \tag{4}$$

Inequality (4) is called a variational inequality and certifies that the feasible flow $\boldsymbol{f}^{(j)}$ is a local optimum, which is also a global optimum, because C^r is convex.

An *optimal offline flow* for MRTW is an optimal solution $\boldsymbol{f}^\star$ of the following convex optimization problem:

$$\begin{aligned} \min \quad & C(\boldsymbol{f}) \\ \text{s.t.} \quad & \sum_{a \in \delta^+(v)} f_a^{(j)} - \sum_{a \in \delta^-(v)} f_a^{(j)} = \gamma_v^{(j)}, && \forall\, v \in V,\, j \in \mathcal{K}_r,\, r \in \mathcal{R} \\ & f_a^{(j)} \geq 0 && \forall\, a \in A,\, j \in \mathcal{K}_r,\, r \in \mathcal{R}. \end{aligned}$$

As in Remark 3.1, an optimal offline solution can be computed in polynomial time within arbitrary precision.

To shorten notation we use the following convention throughout the paper: When we speak of a sequence $\sigma = 1, \dots, R$ of rounds, we refer to the full specification $(d_j, s_j, t_j, \tilde{\tau}_j, T_j)$ of the commodities $j \in \mathcal{K}_r$ for $r = 1, \dots, R$. For a sequence σ, we denote by Opt(σ) the value of an optimal offline flow, and for an online algorithm Alg we denote by Alg(σ) the cost $C(\boldsymbol{f})$ of the solution $\boldsymbol{f}$ produced by Alg for sequence σ.

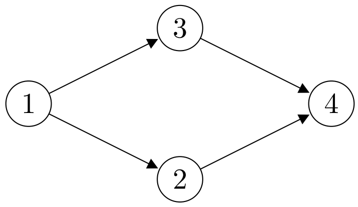

Figure 1: Graph construction for the proof of Propositions 3.3.

3.2. Comparison of Online Algorithms Seq and Seq^2

Before we investigate the competitiveness of Seq and Seq^2, we compare their relative strengths.

Intuitively, algorithm Seq^2 is disadvantaged compared to Seq, since it does not use all available information of a given round r when determining the routing. If there is only one round, Seq computes an optimal solution, and obviously we have that Seq$(\sigma) \leq$ Seq$^2(\sigma)$ in this case. This raises the question whether Seq always outperforms Seq^2?

To study this question, we define the concept of domination: We call an algorithm Alg$_2$ *dominated* by another algorithm Alg$_1$, if we have Alg$_1(\sigma) \leq$ Alg$_2(\sigma)$ for any sequence σ. With this notation, we can answer the above question to the negative:

Proposition 3.3. Seq^2 *is not dominated by* Seq.

Proof. Consider the network in Figure 1. We assume that the two arcs leaving node 1 have price function z and the other two arcs have price function 0. We consider two rounds. The first has two commodities, and the second contains one commodity. All commodities have the same time window of $[0, 1)$.

The first commodity of the first round has a demand of 1 that has to be routed from node 1 to node 4. There are two possible paths for routing this demand. Since the cost on both paths are equal, Seq^2 splits the demand equally. The resulting cost is

$$2 \int_0^1 \int_0^{\frac{1}{2}} z \, dz \, dt = 2 \left[\tfrac{1}{2} z^2\right]_0^{\frac{1}{2}} = \tfrac{1}{4}.$$

The second commodity has a demand of 2, which has to be routed from node 1 to node 3, leaving no routing choice. The cost of Seq^2 for this demand is

$$\int_0^1 \int_0^2 \left(\tfrac{1}{2} + z\right) dz \, dt = \left[\tfrac{1}{2} z + \tfrac{1}{2} z^2\right]_0^2 = 3.$$

Finally, the demand of the second round has a size of 4 and must be routed from node 1 to node 2, again leaving no routing choice. Here, the cost of Seq^2 is $(\frac{1}{2} + \frac{1}{2} \cdot 4) \cdot 4 = 10$. Hence, the total cost for Seq^2 is given by Seq$^2(\sigma) = \frac{1}{4} + 3 + 10 = \frac{53}{4}$; the cost for round 1 is $\frac{13}{4}$.

In contrast, Seq coordinates the routings of the first round in order to minimize the cost. An optimal routing for the two commodities of the first round is to route the first entirely along the lower path $(1, 2, 4)$ and the second on arc $(1, 3)$. This incurs a cost of $\frac{1}{2} + \frac{1}{2} \cdot 2 \cdot 2 = \frac{5}{2}$, which is indeed smaller than the cost of SEQ^2 for this round. For the second round there is no routing choice, and the cost of SEQ for this round is $(1 + \frac{1}{2} \cdot 4) \cdot 4 = 12$. Therefore, the total cost of SEQ is $\frac{5}{2} + 12 = \frac{29}{2}$, which is larger than the cost of $\frac{53}{4}$ obtained by SEQ^2. □

The close relationship between SEQ and SEQ^2 is expressed by the following proposition, which is immediate from the definition of SEQ^2.

Proposition 3.4. *For every instance σ, there exists a sequence σ' such that*

$$\text{SEQ}(\sigma') = \text{SEQ}^2(\sigma) \quad \textit{and} \quad \text{OPT}(\sigma') = \text{OPT}(\sigma).$$

Corollary 3.5.
(1) *Each lower bound on the competitive ratio for* SEQ^2 *is also a lower bound on the competitive ratio for* SEQ.
(2) *Each upper bound on the competitive ratio for* SEQ *is also an upper bound on the competitive ratio for* SEQ^2.

Concluding, SEQ and SEQ^2 do not dominate each other, and if SEQ is c-competitive then SEQ^2 is c-competitive as well. See also the discussion in the conclusions.

4. Competitive Analysis

In Harks et al. [17, 18] it is shown that there exists no competitive deterministic algorithm if the price functions are arbitrary; the used price functions are monomials with a degree tending towards infinity. However, when degrees of the polynomials are bounded, we we derive in the bounded upper bounds on the competitive ratio.

4.1. A General Upper Bound

In this section, we derive a quite general upper bound on the competitive ratio of the online algorithms SEQ and SEQ^2. In the next section we apply this result to polynomials with a given fixed degree.

The general bound is obtained by using the variational inequality in Lemma 3.2 and bounding arc-wise the cost of a flow produced by SEQ with respect to the optimal flow. Then the maximum of these bounds over all arcs is a bound on the competitive ratio. Similar techniques have previously been applied to bounding the price of anarchy in selfish routing problems, see Roughgarden [23] and Correa, Schulz, and Stier-Moses [10]. The bounds on arcs $a \in A$ depend on their price functions $p_a(\cdot)$. We will use the following values for each $a \in A$ and $\lambda \geq 1$:

$$\delta(p_a) := \sup_{x \geq 0} \begin{cases} \dfrac{p_a(x)\, x}{\int_0^x p_a(z)\, dz} & \text{if } \int_0^x p_a(z)\, dz > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

$$\omega(p_a; \lambda) := \sup_{x, f \geq 0} \begin{cases} \dfrac{\big(p_a(f) - \lambda\, p_a(x)\big)\, x}{p_a(f)\, f} & \text{if } p_a(f)\, f > 0 \\ 0 & \text{if } p_a(f)\, f = 0. \end{cases} \tag{6}$$

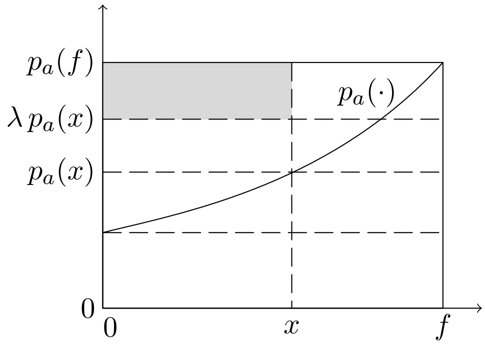

Figure 2: Illustration of the value $\omega(p_a;\lambda)$ in (6) with $1 < \lambda < \frac{p_a(f)}{p_a(x)}$. The gray-shaded area corresponds to the numerator of $\omega(p_a;\lambda)$ and the whole area to the denominator.

Figure 2 gives an illustration of $\omega(p_a;\lambda)$. For a class $\mathcal{C}$ of nondecreasing and continuous price functions, we further define

$$\delta(\mathcal{C}) := \sup_{p_a\in\mathcal{C}} \delta(p_a) \qquad \text{and} \qquad \omega(\mathcal{C};\lambda) := \sup_{p_a\in\mathcal{C}} \omega(p_a;\lambda).$$

The generalized value $\omega(\mathcal{C};\lambda)$ was first introduced by Harks [16] for investigating selfish routing problems involving atomic players. In the same context, a similar value $\beta(\mathcal{C})$, with $\beta(\mathcal{C}) = \omega(\mathcal{C};1)$, was defined in Correa, Schulz, and Stier-Moses [10]. Furthermore, Roughgarden [23] presented the so-called *anarchy value* $\alpha(\mathcal{C})$, with $\alpha(\mathcal{C}) = \big(1-\omega(\mathcal{C};1)\big)^{-1}$.

We define the *feasible scaling set* for $\mathcal{C}$ as

$$\Lambda(\mathcal{C}) := \{\lambda \geq 1 \,:\, 1 - \delta(\mathcal{C})\,\omega(\mathcal{C};\lambda) > 0\}.$$

Theorem 4.1. *If $\Lambda(\mathcal{C}) \neq \varnothing$, then the online algorithms* Seq *and* Seq^2 *are c-competitive, for*

$$c = \inf_{\lambda\in\Lambda(\mathcal{C})} \left[\frac{\lambda\,\delta(\mathcal{C})}{1-\delta(\mathcal{C})\,\omega(\mathcal{C};\lambda)}\right].$$

Proof. Let $\boldsymbol{f}$ be the flow constructed by Seq and $\boldsymbol{x}$ be an arbitrary feasible flow. We want to bound the cost $C(\boldsymbol{f})$ with respect to $C(\boldsymbol{x})$. To this end, we obtain the following sequence of inequalities. We start by applying Lemma 2.1.

$$C(\boldsymbol{f}) = \sum_{r=1}^{R}\sum_{a\in A}\int_{\tau_r}^{T_{r,\max}}\int_{0}^{G_{a,r}(\boldsymbol{f},t)} p_a\big(F_{a,r}(\boldsymbol{f},t)+z\big)\,dz\,dt.$$

Because the price functions are nondecreasing and nonnegative, we have:

$$\begin{aligned} &\leq \sum_{r=1}^{R}\sum_{a\in A}\int_{\tau_r}^{T_{r,\max}} p_a\big(F_{a,r}(\boldsymbol{f},t)+G_{a,r}(\boldsymbol{f},t)\big)\;G_{a,r}(\boldsymbol{f},t)\,dt \\ &= \sum_{r=1}^{R}\sum_{a\in A}\sum_{i=1}^{|E|-1}\int_{\hat{\tau}_i}^{\hat{\tau}_{i+1}} p_a\big(F_{a,r}(\boldsymbol{f},t)+G_{a,r}(\boldsymbol{f},t)\big)\Big(\sum_{j\in J_r(t)} f_a^{(j)}\Big)\,dt, \end{aligned}$$

where we used a decomposition similar to (2) and the definition of $G_{a,r}(\boldsymbol{f}, t)$. Because $J_r(t)$ is constant on each piece, we can reorder terms:

$$= \sum_{r=1}^{R} \sum_{a \in A} \sum_{i=1}^{|E|-1} \sum_{j \in J_r(\hat{\tau}_i)} f_a^{(j)} \int_{\hat{\tau}_i}^{\hat{\tau}_{i+1}} p_a\big(F_{a,r}(\boldsymbol{f}, t) + G_{a,r}(\boldsymbol{f}, t)\big)\, dt.$$

We now use the definition of $J_r(t)$ and undo the decomposition into time segments, to get:

$$= \sum_{r=1}^{R} \sum_{a \in A} \sum_{j \in \mathcal{K}_r} f_a^{(j)} \int_{\tau_r}^{T_j} p_a\big(F_{a,r}(\boldsymbol{f}, t) + G_{a,r}(\boldsymbol{f}, t)\big)\, dt.$$

Using Lemma 3.2 yields:

$$\leq \sum_{r=1}^{R} \sum_{a \in A} \sum_{j \in \mathcal{K}_r} x_a^{(j)} \int_{\tau_r}^{T_j} p_a\big(F_{a,r}(\boldsymbol{f}, t) + G_{a,r}(\boldsymbol{f}, t)\big)\, dt.$$

Arguing in the reverse way as above (by using the definition of $J_r(t)$), we can move $x_a^{(j)}$ into the integral:

$$= \sum_{r=1}^{R} \sum_{a \in A} \int\limits_{\tau_r}^{T_{r,\max}} p_a\big(F_{a,r}(\boldsymbol{f}, t) + G_{a,r}(\boldsymbol{f}, t)\big) \Big(\sum_{j \in J_r(t)} x_a^{(j)} \Big)\, dt.$$

We now use the definitions of $F_{a,r}(\boldsymbol{f}, t)$ and $G_{a,r}(\boldsymbol{f}, t)$ and extend the bounds of the integral (we use that each p_a is nonnegative):

$$\leq \sum_{r=1}^{R} \sum_{a \in A} \int_{0}^{\infty} p_a\Big(\sum_{\ell \in K_r(t)} f_a^{\ell} \Big) \Big(\sum_{j \in J_r(t)} x_a^{(j)} \Big)\, dt.$$

As $J_r(t), K_r(t) \subseteq L(t)$ and $p_a(\cdot)$ is nonnegative and nondecreasing, we can bound this by

$$\leq \sum_{a \in A} \int_{0}^{\infty} p_a\Big(\sum_{j \in L(t)} f_a^{(j)} \Big) \Big(\sum_{j \in L(t)} x_a^{(j)} \Big)\, dt.$$

For $\lambda \geq 1$, we now "add 0" and obtain:

$$\begin{aligned} &= \lambda \sum_{a \in A} \int_{0}^{\infty} p_a\Big(\sum_{j \in L(t)} x_a^{(j)} \Big) \Big(\sum_{j \in L(t)} x_a^{(j)} \Big)\, dt \\ &\quad + \sum_{a \in A} \int_{0}^{\infty} \Big[p_a\Big(\sum_{j \in L(t)} f_a^{(j)} \Big) - \lambda\, p_a\Big(\sum_{j \in L(t)} x_a^{(j)} \Big) \Big] \Big(\sum_{j \in L(t)} x_a^{(j)} \Big)\, dt. \end{aligned}$$

For fixed $a \in A$ and t we use

$$f = \sum_{j \in L(t)} f_a^{(j)} \qquad \text{and} \qquad x = \sum_{j \in L(t)} x_a^{(j)}$$

in the definition of $\delta(p_a)$ and $\omega(p_a;\lambda)$ to get:

$$\leq \lambda \sum_{a\in A} \delta(p_a) \int_0^\infty \int_0^{\sum_{j\in L(t)} x_a^{(j)}} p_a(z)\,dz\,dt + \sum_{a\in A} \omega(p_a;\lambda) \int_0^\infty p_a\Big(\sum_{j\in L(t)} f_a^{(j)}\Big)\Big(\sum_{j\in L(t)} f_a^{(j)}\Big)\,dt.$$

The bound involving $\omega(p_a;\lambda)$ holds because of the following: If $p_a(f)=0$ then trivially $(p_a(f)-\lambda\, p_a(x))\,x \leq 0$ and the bound is true. If $f=0$, then $(p_a(f)-\lambda p_a(x)) \leq 0$ since $\lambda \geq 1$ and $p_a(\cdot)$ is nondecreasing. The case $p_a(f)\,f>0$ follows from the definition.

Using the definition of $\delta(\mathcal{C})$ and applying $\delta(p_a)$ to the second term yields:

$$\begin{aligned} &\leq \lambda\,\delta(\mathcal{C})\;C(\boldsymbol{x}) + \sum_{a\in A} \omega(p_a;\lambda)\;\delta(p_a) \int_0^\infty \int_0^{\sum_{j\in L(t)} f_a^{(j)}} p_a(z)\,dz\,dt \\ &\leq \lambda\,\delta(\mathcal{C})\;C(\boldsymbol{x}) + \delta(\mathcal{C})\;\omega(\mathcal{C};\lambda)\;C(\boldsymbol{f}). \end{aligned}$$

Applying this inequality to the optimal offline solution $\boldsymbol{x}$, rewriting, and taking the infimum over $\lambda \in \Lambda(\mathcal{C})$ yields the desired result for Seq. The result for Seq^2 follows from Corollary 3.5 (2). □

4.2. Polynomial Price Functions

To facilitate the result of Theorem 4.1, one needs to bound $\delta(\mathcal{C})$ and $\omega(\mathcal{C};\lambda)$. In this section we consider the class $\mathcal{C}_d$ of polynomials with nonnegative coefficients and degree at most $d \in \mathbb{N}$:

$$\mathcal{C}_d := \{c_d\,x^d + \cdots + c_1\,x + c_0 \;:\; c_s \geq 0,\; s=0,\ldots,d\}.$$

Note that polynomials in $\mathcal{C}_d$ are nonnegative for nonnegative arguments, continuous, nondecreasing, and convex.

Theorem 4.2. *For price functions in $\mathcal{C}_d$, the online algorithms* Seq *and* Seq^2 *are $(d+1)^{d+1}$-competitive.*

Before we prove Theorem 4.2, we first derive a bound on the value $\delta(\mathcal{C}_d)$ and $\omega(\mathcal{C}_d;\lambda)$.

Lemma 4.3. *For the class $\mathcal{C}_d$ of polynomials with maximal degree d, the value $\delta(\mathcal{C}_d)$ is at most $d+1$.*

Proof. Let $p_a(x) = c_d\,x^d + \cdots + c_1\,x + c_0 \in \mathcal{C}_d$. We apply the definition of the value $\delta(p_a)$:

$$\delta(p_a) = \sup_{x\geq 0} \frac{x \sum_{i=0}^{d} c_i\,x^i}{\sum_{i=0}^{d} \frac{c_i}{i+1} x^{i+1}} \leq \sup_{x\geq 0} \frac{\sum_{i=0}^{d} c_i\,x^{i+1}}{\sum_{i=0}^{d} \frac{c_i}{d+1} x^{i+1}} = \frac{1}{\frac{1}{d+1}} = d+1,$$

where the inequality follows since $c_i \geq 0$ and $x \geq 0$. □

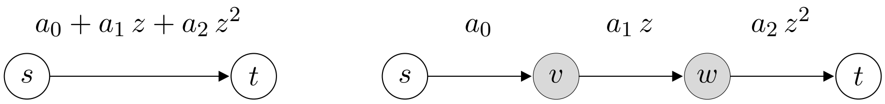

Figure 3: Reduction of polynomials to monomials. By introducing the two nodes v and w, the arc (s, t) is partitioned into three separate arcs with monomial price functions.

We now observe that the cost function $C(\boldsymbol{f})$ is linear in each of the price functions $p_a(\cdot)$. We can therefore reduce the analysis to monomial price functions. For this we subdivide each arc a into d arcs $a_1, \dots, a_d$ with monomial price functions $p_{a_s}(x) = c_s\, x^s$ for any $s = 1, \dots, d$, see Figure 3.

The following lemma can also be found in Harks [16]. We present a proof for completeness.

Lemma 4.4. *For the class $\mathcal{C}$ of monomials $c_s\, x^s$ of degree $s \in \{1, \dots, d\}$ with $c_s \geq 0$ and for $\lambda \geq 1$, we have*

$$\omega(\mathcal{C}; \lambda) \leq \max_{0 \leq \mu} \big(\mu - \lambda\, \mu^{d+1}\big).$$

Proof. For $p_a(\cdot) \in \mathcal{C}$, we can assume that $p_a(f)\, f > 0$, since otherwise $\omega(\mathcal{C}; \lambda) = 0$ and the claim is trivially true. By definition, we have

$$\omega(p_a; \lambda) = \sup_{x, f \geq 0} \frac{\big(p_a(f) - \lambda\, p_a(x)\big)\, x}{p_a(f)\, f}.$$

Defining $\mu := \frac{x}{f}$ (recall that $f > 0$), we obtain

$$\omega(p_a; \lambda) = \sup_{0 \leq \mu} \frac{\big(p_a(f) - \lambda\, p_a(\mu\, f)\big)\, \mu\, f}{p_a(f)\, f}.$$

Consider the monomial price function $p_a(x) = c_s\, x^s$ of degree $s \in \{1, \dots, d\}$. To bound the value $\omega(p_a; \lambda)$ from above, we have to compute:

$$\omega(p_a; \lambda) = \sup_{0 \leq \mu} \frac{(c_s\, f^s - \lambda\, c_s\, \mu^s\, f^s)\, \mu\, f}{c_s\, f^{s+1}} = \max_{0 \leq \mu} \big(\mu - \lambda\, \mu^{s+1}\big). \tag{7}$$

Because of the assumption $\lambda \geq 1$, the maximum is attained at a point with $\mu \leq 1$. It follows that

$$\max_{0 \leq \mu} \big(\mu - \lambda\, \mu^{s+1}\big) \leq \max_{0 \leq \mu} \big(\mu - \lambda\, \mu^{d+1}\big).$$

This shows the claim. □

For polynomials in $\mathcal{C}_d$ and an appropriate choice of λ, we can prove the following bound on $\omega(\mathcal{C}_d; \lambda)$.

Proposition 4.5. *For $\lambda := (d+1)^{(d-1)} \geq 1$, we have $\omega(\mathcal{C}_d; \lambda) \leq \frac{d}{(d+1)^2}$.*

Proof. Using Lemma 4.4 we get

$$\omega(\mathcal{C}_d; \lambda) \leq \max_{0 \leq \mu} \big(\mu - \lambda\, \mu^{d+1}\big) = \max_{0 \leq \mu} \big(\mu - (d+1)^{(d-1)}\, \mu^{d+1}\big).$$

The unique solution turns out to be $\mu^\star = \frac{1}{d+1}$. Evaluating the objective leads to:

$$\omega(\mathcal{C}_d; \lambda) \leq \frac{1}{d+1} - (d+1)^{(d-1)} \Big(\frac{1}{d+1}\Big)^{d+1} = \frac{d}{(d+1)^2}.$$

This proves the claim. □

With these prerequisites we can prove Theorem 4.2.

Proof of Theorem 4.2. Let the flow $\boldsymbol{f}$ be produced by the online algorithm SEQ and let $\boldsymbol{x}$ be an arbitrary feasible flow for the given instance. We define $\lambda := (d+1)^{(d-1)}$ and apply Proposition 4.5, which yields $\omega(\mathcal{C}_d;\lambda) \leq \frac{d}{(d+1)^2}$. In order to apply Theorem 4.1, we have to verify that $\lambda \in \Lambda(\mathcal{C}_d)$. What remains to be shown is that

$$1-(d+1)\frac{d}{(d+1)^2} > 0$$

holds, where $\delta(\mathcal{C}_d) \leq d+1$ by Lemma 4.3. This inequality is equivalent to $\frac{1}{d+1} > 0$, which is trivially true. Applying Theorem 4.1 yields

$$C(\boldsymbol{f}) \leq \frac{(d+1)^{(d-1)}\,(d+1)}{\left(1-(d+1)\frac{d}{(d+1)^2}\right)}\,C(\boldsymbol{x}) = (d+1)^{d+1}\,C(\boldsymbol{x}).$$

Taking $\boldsymbol{x}$ as the optimal offline solution proves the claim. By Corollary 3.5 (2) the result for SEQ^2 follows as well. □

Corollary 4.6. *For affine linear price functions $p_a(\cdot)$ with nonnegative coefficients, i.e., $p_a(\cdot) \in \mathcal{C}_1$,* SEQ *and* SEQ^2 *are 4-competitive.*

For affine linear price functions, Harks et al. [18] proved that SEQ is $(4n^2)/(1+n)^2$-competitive when there are no time windows and there is only one commodity per round. The analysis, however, is tailored to affine linear price functions and does not provide bounds for higher degree polynomials. We do not know whether this result can be generalized to improve the bound of Corollary 4.6 (made commodity dependent).

Corollary 4.7. *For concave price functions, the competitive ratio of* SEQ *and* SEQ^2 *is bounded by 4.*

Proof. The statement follows from a geometrical argument similar to Correa et al. [9], see Figure 2 for an illustration. The worst case ratio between the two areas in the figure occurs exactly in the case in which the price function is linear. □

4.3. Resource Augmentation

As noted above, in general, no online algorithm is competitive if the set of allowable price functions is not restricted. This may be a sign that the adversary (offline optimum) is too powerful. To limit the power of the adversary, we can give her the disadvantage that she has to route a demand that is increased by a factor $\gamma \geq 1$. This can be viewed as a way of resource augmentation, see e.g., Kalyanasundaram and Pruhs [20] or Roughgarden and Tardos [23].

Before we state an upper bound in this context, we generalize the definition of $\Lambda(\mathcal{C})$ as follows.

$$\Lambda(\mathcal{C};\gamma) := \{\lambda \geq 1 \,:\, \gamma - \delta(\mathcal{C})\,\omega(\mathcal{C};\lambda) > 0\}.$$

Theorem 4.8. *Let the price functions on the arcs be from a class of non-decreasing and continuous functions $\mathcal{C}$, for which $\Lambda(\mathcal{C};\gamma) \neq \varnothing$, and suppose that the adversary* ADV *needs to route a demand increased by a factor $\gamma \geq 1$. Then for any request sequence σ, we have that* $\text{SEQ}(\sigma) \leq c \cdot \text{ADV}(\sigma)$, *where*

$$c = \inf_{\lambda \in \Lambda(\mathcal{C};\gamma)} \left[\frac{\lambda\, \delta(\mathcal{C})}{\gamma - \delta(\mathcal{C})\, \omega(\mathcal{C};\lambda)} \right].$$

Proof. Let $\boldsymbol{x}$ be the optimal flow with respect to γ. Using Lemma 3.2 we obtain for commodity j:

$$\sum_{a \in A} \Big[\int_{\tau_r}^{T_j} p_a\big(F_{a,r}(\boldsymbol{f},t) + G_{a,r}(\boldsymbol{f},t)\big)\, dt \Big] \big(f_a^{(j)} - \frac{x_a^{(j)}}{\gamma}\big) \leq 0,$$

since $\frac{x_a^{(j)}}{\gamma}$ is a feasible flow for commodity j. Following the proof of Theorem 4.1, we get:

$$\gamma\, C(\boldsymbol{f}) \leq \ \lambda\, \delta(\mathcal{C})\ C(\boldsymbol{x}^s tar) + \delta(\mathcal{C})\ \omega(\mathcal{C};\lambda)\ C(\boldsymbol{f}).$$

Rewriting proves the claim. □

If we restrict the set of allowable price functions to polynomials in $\mathcal{C}_d$, we can show the following:

Corollary 4.9. *Let the price functions on the arcs be in $\mathcal{C}_d$, and suppose that the adversary* ADV *needs to route a factor $\gamma \geq 1$ as much demand per request than the online algorithm* SEQ. *Then, for any request sequence σ, we have that*

$$\text{SEQ}(\sigma) \leq \frac{(d+1)^{d+1}}{(d+1)\,\gamma - d}\, \text{ADV}(\sigma).$$

Proof. Defining $\lambda := (d+1)^{d-1}$ and applying Lemma 4.5, Lemma 4.3, and Theorem 4.8 we get:

$$\gamma\, C(\boldsymbol{f}) \leq \ (d+1)^d\ C(\boldsymbol{x}) + \frac{d}{d+1}\ C(\boldsymbol{f}).$$

Rewriting proves the claim. □

Note that for $\gamma = (d+1)^d + \frac{d}{d+1}$ we have $\text{SEQ}(\sigma) \leq \text{ADV}(\sigma)$. In particular, this implies that for affine linear price functions, the cost of SEQ for a given sequence σ does not exceed the cost of ADV with respect to $\gamma = 2.5$.

Using Corollary 3.5 (2), the statement of Theorem 4.8 and Corollary 4.9 hold for SEQ^2 as well.

4.4. Relation to the Load Balancing Problem

In the following, we will obtain lower bounds for the online MRTW by transfering lower bounds for the *online load balancing problem* (OLB). In this scheduling problem, jobs arrive in an online fashion and have to be scheduled on machines in order to minimize the ℓ_d-Norm of the vector of the server loads. Each job can only be scheduled on a given subset of permissible machines. It turns out that our setting generalizes the special case of *related* (parallel) machines: each job induces work of w_j when it is scheduled on any permissable machine. This problem has received much attention, see

jobs	perm. machines
1	1, 2
2	2
3	3, 4

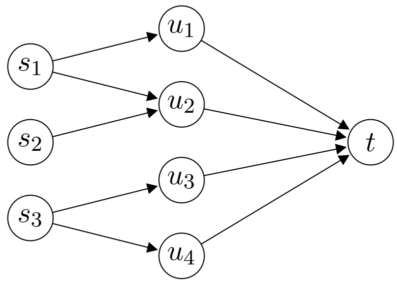

Figure 4: Example of the graph construction of Section 4.4.

Awerbuch et al. [4], Caragiannis et al. [8], and Suri et al. [25]. In the following we will describe the relation between this problem and our setting.

Given are J independent jobs, each of which needs to be scheduled on one of M parallel machines. The jobs are released in an online manner, such that job j arrives before job $j+1$, for $j = 1, \dots, J-1$. Each job j can be scheduled on a subset of permissible machines on which it induces work w_j. That is, each job j induces work of $w_{ij} \in \{w_j, \infty\}$ on machine $i \in \{1, \dots, M\}$. Each machine i has speed q_i and the *load* ℓ_i of machine i is the sum of weights w_j of jobs assigned to it divided by its speed q_i.

A *schedule* is determined by a matrix $S = (S_{ij})$ with $S_{ij} \in [0, w_j]$, for $i \in \{1, \dots, M\}$, $j \in \{1, \dots, J\}$. A schedule is *feasible* if

$$\sum_{i=1}^{M} S_{ij} = w_j,$$

and $S_{ij} = 0$ if $w_{ij} = \infty$. For the unsplittable variant it is required that $S_{ij} \in \{0, w_{ij}\}$. The cost of a schedule S is defined as

$$L(S) := \Big(\sum_{i=1}^{M} \ell_i^d\Big)^{1/d} = \Big(\sum_{i=1}^{M} \Big(\frac{1}{q_i} \sum_{j=1}^{J} S_{ij}\Big)^d\Big)^{1/d},$$

for $d \in \mathbb{N}$, which is the the ℓ_d-norm of the server loads. We call the tuple $\sigma = (M, \boldsymbol{q}, J, \boldsymbol{w}, d)$ an instance of OLB.

Suppose we are given an instance σ for OLB. Then we construct an equivalent instance $\mathcal{G}(\sigma)$ (consisting of a directed graph D with commodities and price functions) for MRTW. The graph D contains nodes s_j for each job j. Furthermore, for each machine i it contains nodes u_i and one additional node t. There are arcs (u_i, t) for all machines i. For each job j, the set of permissible machines define a set of arcs $A_j := \{(s_j, u_i) : w_{ij} \neq \infty\}$. We define $A := A_1 \cup \dots \cup A_J$. See Figure 4 for an example. For every arc $a = (u_i, t)$ the price function is $p_a(x) = d \cdot (x/q_i)^{d-1}$. All remaining arcs have constant price functions 0.

We define a sequence of rounds $\sigma' = 1, \dots, J$, where each round j contains one commodity with demand w_j that has to be routed from s_j to t. We do not consider time windows, i.e., we set $\tilde{\tau}_j = 0$, $T_j = 1$, see Remarks 2.2 and 2.3. This completes the specification of the instance $\mathcal{G}(\sigma)$.

By flow conservation, a feasible flow $\boldsymbol{f}$ for $\mathcal{G}(\sigma)$ is completely determined by the flow values f_a^j, $a = (s_j, u_i) \in A$. We let

$$S_{ij}(\boldsymbol{f}) := \begin{cases} f_a^j & \text{for } a = (s_j, u_i) \in A \\ 0 & \text{otherwise} \end{cases}$$

be the schedule $S(\boldsymbol{f})$ *induced by flow* $\boldsymbol{f}$. Conversely, we say that flow $\boldsymbol{f}(S)$ is induced by the schedule S if $f_a^j(S) := S_{ij}$, for $(s_j, u_i) \in A$, and extended by flow conservation to the remaining arcs. Note that this transformation works in an online fashion since it can be done commodity (job) wise. We obtain the following straightforward observation.

Lemma 4.10. *Let σ be an instance for* OLB*. Then, every feasible schedule S for σ induces a feasible flow $\boldsymbol{f}(S)$ for $\mathcal{G}(\sigma)$, and every feasible flow $\boldsymbol{f}$ for $\mathcal{G}(\sigma)$ induces a feasible schedule $S(\boldsymbol{f})$ for σ.*

The relation of the corresponding objective function values is as follows.

Lemma 4.11. *Let σ be an instance for* OLB*. Then, every feasible schedule S for σ and feasible flow $\boldsymbol{f}$ for $\mathcal{G}(\sigma)$ satisfy*

$$L(S) = C(\boldsymbol{f}(S))^{1/d} \qquad \textit{and} \qquad C(\boldsymbol{f}) = L(S(\boldsymbol{f}))^d.$$

Proof. Let S be any feasible schedule for σ with cost $L(S)$. The cost of flow $\boldsymbol{f}(S)$ evaluates as follows:

$$\begin{aligned} C(\boldsymbol{f}(S)) &= \sum_{i=1}^{M} \int_0^{\sum_{j=1}^J f^j_{(u_i,t)}} d \cdot (z/q_i)^{d-1}\, dz \\ &= \sum_{i=1}^{M} \int_0^{\sum_{j=1}^J f^j_{(s_j,u_i)}} d \cdot (z/q_i)^{d-1}\, dz \\ &= \sum_{i=1}^{M} \int_0^{\sum_{j=1}^J S_{ij}} d \cdot (z/q_i)^{d-1}\, dz \\ &= \sum_{i=1}^{M} \big(\frac{1}{q_i} \sum_{j=1}^{J} S_{ij}\big)^d = L(S)^d. \end{aligned}$$

The reverse direction follows similarly. □

Corollary 4.12. *Let σ be an instance for* OLB*. Then, every optimal solution $S^\star$ for σ induces an optimal solution $\boldsymbol{f}(S^\star)$ for $\mathcal{G}(\sigma)$. Conversely, every optimal solution $\boldsymbol{f}^\star$ for $\mathcal{G}(\sigma)$ induces an optimal solution $S(\boldsymbol{f}^\star(S))$ for σ.*

Proof. Let $S^\star$ be an optimal schedule for σ. By Lemma 4.11, we know that $S^\star$ induces a feasible flow $\boldsymbol{f}(S^\star)$ with $L(S^\star) = C(\boldsymbol{f}(S^\star))^{1/d}$. Assume $\boldsymbol{f}(S^\star)$ is not optimal for $\mathcal{G}(\sigma)$. Then, there exists a feasible flow $\boldsymbol{x}$ with $C(\boldsymbol{x}) < C(\boldsymbol{f}(S^\star))$. Then, $\boldsymbol{x}$ induces a feasible schedule $S(\boldsymbol{x})$ with cost $L(S(\boldsymbol{x})) = C(\boldsymbol{x})^{1/d} < C(\boldsymbol{f}(S^\star))^{1/d} = L(S^\star)$, contradicting the optimality of $S^\star$. The reverse direction follows similarly. □

For polynomial price functions, we obtain:

Lemma 4.13. *If* ALG *is a c-competitive deterministic online algorithm for the online* MRTW *with price functions in $\mathcal{C}_{d-1}$, then there exist a $c^{1/d}$-competitive deterministic online algorithm* ALG′ *for* OLB.

Proof. Suppose ALG is a c-competitive deterministic online algorithm for MRTW with price functions in $\mathcal{C}_{d-1}$. Then, for any given instance σ of OLB we construct the instance $\mathcal{G}(\sigma)$, which has price functions in $\mathcal{C}_{d-1}$. Let $\boldsymbol{f}$ be the flow computed by ALG on $\mathcal{G}(\sigma)$. Algorithm ALG′ is the deterministic online algorithm that returns the induced schedule $S(\boldsymbol{f})$ for σ. By Lemma 4.11, it follows that $L(S(\boldsymbol{f})) = C(\boldsymbol{f})^{1/d}$. Combining Corollary 4.12 and Lemma 4.11 we know that $L(S^\star) = C(\boldsymbol{f}^\star)^{1/d}$ for optimal solutions $S^\star$ of σ and $\boldsymbol{f}^\star$ of $\mathcal{G}(\sigma)$. By assumption we have that $C(\boldsymbol{f}) \leq c \cdot C(\boldsymbol{f}^\star)$. Thus, it follows that $L(S(\boldsymbol{f})) \leq c^{1/d} \cdot L(S^\star)$. Since the input instance σ was arbitrary, the claim is proven. ☐

Theorem 4.14. *A lower bound c for the competitive ratio of any deterministic online algorithm for* OLB *with ℓ_d-norm carries over to a lower bound of c^d for the competitive ratio of any deterministic online algorithm for the online* MRTW *with price functions in $\mathcal{C}_{d-1}$.*

Proof. Let c be a lower bound for the competitive ratio of any deterministic online algorithm for OLB. Assume there exist a $\hat{c}$-competitive deterministic online algorithm for the online MRTW with $\hat{c} < c^d$. Then, Lemma 4.13 implies that we can construct an online algorithm ALG′, which is $\hat{c}^{1/d}$-competitive for OLB. Monotonicity of the d-th root implies $\hat{c}^{1/d} < c$ contradicting the first statement. ☐

We now generalize the above setting in two ways. First, we can easily include *time windows.* Second, we can consider online load balancing problems in the ℓ_d-norm for *general networks.* Hence, the cost of a flow $\boldsymbol{f}$ (generalized schedule) is

$$L(\boldsymbol{f}) := \Big[\sum_{a\in A} \int_0^\infty \Big(\sum_{j\in L(t)} f_a^{(j)}\Big)^d \, dt\Big]^{1/d}.$$

Corollary 4.15. *For the online load balancing problem in general networks, time windows, and the ℓ_d-norm,* SEQ *and* SEQ^2 *are d-competitive. Furthermore, there exist an instance σ with* $\text{SEQ}^2(\sigma) \geq d \cdot \text{OPT}(\sigma)$.

Proof. Using Theorem 4.1 for price functions in $\mathcal{C}_{d-1}$, we get

$$L(\boldsymbol{f})^d = \sum_{a\in A} \int_0^\infty \int_0^{\sum_{j\in L(t)} f_a^{(j)}} d\cdot z^{d-1}\, dz\, dt = C(\boldsymbol{f}) \leq d^d\, C(\boldsymbol{x}) = d^d\, L(\boldsymbol{x})^d.$$

A lower bound can be constructed based on a construction in Farzad et al. [12], see Theorem 4.18 below. ☐

4.5. Lower Bounds on the Competitive Ratio

Before we give lower bounds on the competitive ratio of the online algorithms SEQ and SEQ^2, we present a general lower bound for *all* online algorithms. We again consider the class of nonnegative and nondecreasing polynomial price functions $\mathcal{C}_d$ for $d \in \mathbb{N}$ and use an instance of the OLB to derive a lower bound (see Theorem 4.14). The construction of this bound is based on ideas of Awerbuch et al. [4]. The work in [4], however, only deals with unsplittable jobs, whereas the following bound holds for the splittable case.

Theorem 4.16. *For price functions in $\mathcal{C}_d$, the competitive ratio of any deterministic online algorithm for the online* MRTW *is at least* $(0.265\,(d{+}1))^{d+1}$.

Proof. Let ALG be a deterministic online algorithm for the OLB. We are given $M = 2^k$ parallel machines and construct the input sequence σ of jobs depending on the decisions of the online algorithm. The machines will be either labeled as *active* or *inactive* at each step of the sequence. Initially all machines are active.

There are k rounds. In each round, the active machines are grouped into pairs. For each pair of machines (i, i'), the sequence σ contains a job j of weight $w_j = 1$, whose permissible machines are i and i'. Assume that ALG produces a schedule S by placing $\alpha \in [0, 1]$ units on machine i and $1 - \alpha$ on machine i'. The sequence is constructed in such a way that for every job the machine with the smaller fraction is labeled inactive (ties are broken arbitrarily). For bounding the cost of ALG from below we can assume $\alpha = \frac{1}{2}$.

In each new round only half of the machines of the previous round are active. We construct a schedule $S^\star$ by placing a job entirely on the machine that becomes inactive after the corresponding round. The cost with respect to the ℓ_d-norm of $S^\star$ is

$$L(S^\star)^d = 2^{k-1} + 2^{k-2} + \cdots + 2 + 1 = 2^k - 1.$$

The cost of the schedule S of ALG is given by

$$L(S)^d = \sum_{i=1}^{k} \left(\frac{i}{2}\right)^d 2^{k-i}.$$

The ratio of the cost of S and $S^\star$ is bounded from below by

$$\frac{L(S)^d}{L(S^\star)^d} = \frac{\sum_{i=1}^{k} (\frac{i}{2})^d\, 2^{k-i}}{2^k - 1} \geq \left(\frac{\log_2(e)\, d}{2\, e}\right)^d \geq (0.265 \cdot d)^d.$$

As Theorem 4.14 considers price functions in $\mathcal{C}_{d-1}$, we can replace d by $d{+}1$, obtaining the desired result. □

Remark 4.17. For $d = 1$, Harks et al. [17, 18] proved a lower bound of $\frac{4}{3}$ for any deterministic online algorithm, which in this case is better than the above bound.

Theorem 4.18. *The online algorithms* SEQ *and* SEQ^2 *have a competitive ratio of at least* $(d + 1)^{d+1}$*, even without time windows and only one commodity per round.*

Proof. Farzad et al. [12] presented an example that, when translated to the online MRTW (without time windows and only one commodity per round), provides the lower bound. □

The networks used in the proofs of the previous theorems have the property that the commodities have different sources and destinations. In Harks et al. [17, 18], it is shown that SEQ (or SEQ^2) compute an optimal routing for

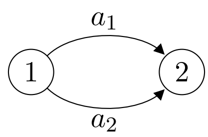

Figure 5: Graph construction for the proof of Proposition 4.19.

networks consisting of two nodes connected by parallel arcs, time windows of size $[0,1]$, and arbitrary nondecreasing price functions. This raises the question whether SEQ (or SEQ^2) returns an optimal solution for parallel arcs and arbitrary time windows. The following theorem shows that this is not true.

Theorem 4.19. *If the input graph consists of two nodes connected by parallel arcs,* SEQ *and* SEQ^2 *cannot be better than* $(2-\frac{1}{2}\sqrt{3})$*-competitive, even for affine linear price functions.*

Proof. Consider the network depicted in Figure 5. Each commodity has node 1 as source and node 2 as destination. Further, arc a_1 has a constant price function of $p_{a_1}(z)=1$ and a_2 a linear price function of $p_{a_2}(z)=z$.

The request sequence consists of two rounds. The first round contains one commodity with demand 1 and time window $[0,1)$. The second round contains one commodity with demand 1 and time window $[0,T)$, for $T=1+\sqrt{3}$.

The cost for routing the first commodity completely over arc a_1 is 1 and over arc a_2 it is $\frac{1}{2}$. Therefore, SEQ^2 assigns the demand of the first commodity completely to arc a_2.

Suppose we route an amount of $\alpha \in [0,1]$ of the demand of the second commodity on arc a_2 and $1-\alpha$ over arc a_1. Then the total cost for the second commodity is

$$\int_0^T \int_0^{1-\alpha} dz\,dt + \int_0^1 \int_0^{\alpha} (1+z)\,dz\,dt + \int_1^T \int_0^{\alpha} z\,dz\,dt = \tfrac{1}{2}\,T\alpha^2 - (T-1)\,\alpha + T. \tag{8}$$

By definition, SEQ (and SEQ^2) computes a feasible flow for the second commodity that minimizes (8). That is, it sends a flow of $\alpha = \frac{T-1}{T}$ on arc a_2 and the rest over arc a_1. The total cost for this solution is

$$\frac{1}{2} + \frac{(T-1)^2}{2\,T} - \frac{(T-1)^2}{T} + T = \frac{1}{2}\,T + \frac{3}{2} - \frac{1}{2\,T}.$$

On the other hand, if we route the first commodity completely over a_1 and the second commodity fully over a_2, we obtain a solution with cost

$$\int_0^1 \int_0^1 dz\,dt + \int_0^T \int_0^1 z\,dz\,dt = 1 + \tfrac{1}{2}\,T.$$

Hence, the competitive ratio cannot be better than

$$\frac{\frac{T}{2} + \frac{3}{2} - \frac{1}{2\,T}}{\frac{T}{2} + 1} = 2 - \frac{1}{2}\sqrt{3},$$

for $T = 1+\sqrt{3}$, which concludes the proof. □

The above theorem shows that the introduction of time windows has an effect on the performance of the online algorithms Seq and Seq^2.

5. Unsplittable Routings

In this section, we study the *unsplittable* MRTW: the variant of MRTW in which the demand of each commodity has to be routed along a single path. For this variant, it will turn out that the obtained bounds are larger than those for the splittable version. Furthermore, whereas in the splittable version the solutions to the offline problem and Seq can be computed in polynomial time (within arbitrary precision), this is not true for the corresponding unsplittable counterparts unless P = NP.

In the following, we study the unsplittable versions of Seq and Seq^2, which we call U-Seq and U-Seq^2, respectively. For each round, U-Seq greedily computes an unsplittable routing, minimizing the cost for the corresponding demands. More formally, for each round $r \in \mathcal{R}$ and given unsplittable flows $\boldsymbol{f}^1, \dots, \boldsymbol{f}^{r-1}$ of the previous rounds, U-Seq minimizes $C^r(\boldsymbol{f}^r; \boldsymbol{f}^1, \dots, \boldsymbol{f}^{r-1})$ to obtain the unsplittable flow $\boldsymbol{f}^r$.

The online algorithm U-Seq^2 routes the demands of a round one by one along the cheapest path with respect to all previously routed demands.

Remark 5.1. As shown in Harks et al. [17], the offline problem of MRTW is NP-hard, even without time windows. It follows that the problem to minimize $C^r(\boldsymbol{f}^r)$, which U-Seq has to solve, is also NP-hard. However, the online algorithm U-Seq^2 has polynomial running time, since it only has to solve a shortest path problem for every commodity of a given round.

The same argumentation of Corollary 3.5 applies to U-Seq and U-Seq^2 and yields:

Corollary 5.2.

1. *Each lower bound on the competitive ratio for* U-Seq^2 *is also a lower bound on the competitive ratio for* U-Seq.
2. *Each upper bound on the competitive ratio for* U-Seq *is also an upper bound on the competitive ratio for* U-Seq^2.

In the following lemma, we express the cost minimization of U-Seq associated with every round r, given the routing decisions of the previous rounds.

Lemma 5.3. *Let $\boldsymbol{f}^1, \dots, \boldsymbol{f}^{r-1}$ be feasible flows for the commodities of the first $r-1$ rounds. If a feasible flow $\boldsymbol{f}^r = (\boldsymbol{f}^{(j)} : j \in \mathcal{K}_r)$ for round r is computed by* U-Seq, *then the following inequality holds for any other feasible flow $\boldsymbol{x}^r = (\boldsymbol{x}^{(j)} : j \in \mathcal{K}_r)$ for round r.*

$$\sum_{a \in A} \int_{\tau_r}^{T_{r,\max}} \int_{0}^{G_{a,r}(\boldsymbol{f},t)} p_a\big(F_{a,r}(\boldsymbol{f},t) + z\big)\, dz\, dt \tag{9}$$

$$\leq \sum_{a \in A} \int_{\tau_r}^{T_{r,\max}} \int_{0}^{G_{a,r}(\boldsymbol{x},t)} p_a\big(F_{a,r}(\boldsymbol{f},t) + z\big)\, dz\, dt. \tag{10}$$

Proof. The inequality simply follows from the definition of U-Seq routing the demands of round r with minimum cost. □

5.1. Upper Bounds for the Unsplittable MRTW

Using similar techniques as in the proof of Theorem 4.1, we define for a given price function $p(z)$, $\lambda \geq 1$, and nonnegative values f, x, the following values.

$$\bar{\omega}(p_a;\lambda) := \sup_{x,f\geq 0} \begin{cases} \dfrac{\int_0^x p_a(f+z)\,dz - \lambda\, p_a(x)\,x}{p_a(f)\,f} & \text{if } p_a(f)\,f > 0 \\ 0 & \text{if } p_a(f)\,f = 0. \end{cases} \tag{11}$$

For a class $\mathcal{C}$ of nondecreasing price functions, we further define

$$\bar{\omega}(\mathcal{C};\lambda) := \sup_{p_a\in\mathcal{C}} \bar{\omega}(p_a;\lambda).$$

Definition 5.4. For a given class of price functions $\mathcal{C}$, the *feasible scaling set* for λ is defined as

$$\Lambda(\mathcal{C}) = \Lambda_1(\mathcal{C}) \cap \Lambda_2(\mathcal{C}),$$

where

$$\Lambda_1(\mathcal{C}) := \{\lambda \geq 1 \,:\, 1 - \bar{\omega}(\mathcal{C};\lambda) > 0\}$$

and

$$\Lambda_2(\mathcal{C}) := \{\lambda \geq 1 \,:\, \int_0^x p_a(f+z)\,dz - \lambda\, p_a(x)\,x \leq 0 \;\text{ for all } f, x \in \mathbb{R}^+ \text{ and } p_a \in \mathcal{C} \text{ with } p_a(f) = 0\}.$$

Theorem 5.5. *If $\Lambda(\mathcal{C}) \neq \varnothing$, the online algorithms* U-Seq *and* U-Seq^2 *are c-competitive for the online* MRTW*, with*

$$c = \inf_{\lambda\in\Lambda(\mathcal{C})} \left[\frac{\lambda\,\delta(\mathcal{C})}{1-\delta(\mathcal{C})\,\bar{\omega}(\mathcal{C};\lambda)}\right].$$

Proof. Let $\boldsymbol{f}$ be the flow constructed by U-Seq and $\boldsymbol{x}$ be an arbitrary feasible flow for the unsplittable MRTW. We want to bound the cost $C(\boldsymbol{f})$ with respect to $C(\boldsymbol{x})$. To this end, we obtain the following sequence of inequalities. We start by applying Lemma 5.3.

$$\begin{aligned} C(\boldsymbol{f}) &= \sum_{r=1}^{R}\sum_{a\in A} \int_{\tau_r}^{T_{r,\max}} \int_0^{G_{a,r}(\boldsymbol{f},t)} p_a\big(F_{a,r}(\boldsymbol{f},t)+z\big)\,dz\,dt \\ &\leq \sum_{r=1}^{R}\sum_{a\in A} \int_{\tau_r}^{T_{r,\max}} \int_0^{G_{a,r}(\boldsymbol{x},t)} p_a\big(F_{a,r}(\boldsymbol{f},t)+z\big)\,dz\,dt. \end{aligned}$$

We now extend the bounds of the integral (we use that each p_a is nonnegative and nondecreasing) and use similar arguments as in the proof of Theorem 4.1 and obtain:

$$\leq \sum_{a\in A} \int_0^{\infty} \int_0^{\sum_{j\in L(t)} x_a^{(j)}} p_a\big(\sum_{j\in L(t)} f_a^{(j)} + z\big)\,dz\,dt.$$

For $\lambda \geq 1$, we now "add 0" and obtain:

$$= \lambda \sum_{a\in A} \int_0^\infty p_a\Big(\sum_{j\in L(t)} x_a^{(j)}\Big)\Big(\sum_{j\in L(t)} x_a^{(j)}\Big)\,dt$$

$$+\sum_{a\in A}\int_0^\infty \int_0^{\sum_{j\in L(t)} x_a^{(j)}} p_a\big(\sum_{j\in L(t)} f_a^{(j)} + z\big)\,dz - \lambda\, p_a\Big(\sum_{j\in L(t)} x_a^{(j)}\Big)\Big(\sum_{j\in L(t)} x_a^{(j)}\Big)\,dt.$$

For fixed $a \in A$ and t we use

$$f := \sum_{j\in L(t)} f_a^{(j)} \qquad \text{and} \qquad x := \sum_{j\in L(t)} x_a^{(j)}$$

in the definition of $\delta(p_a)$ and $\bar\omega(p_a;\lambda)$ to get:

$$\leq \lambda \sum_{a\in A} \delta(p_a) \int_0^\infty \int_0^{\sum_{j\in L(t)} x_a^{(j)}} p_a(z)\,dz\,dt$$

$$+ \sum_{a\in A} \bar\omega(p_a;\lambda) \int_0^\infty p_a\Big(\sum_{j\in L(t)} f_a^{(j)}\Big)\Big(\sum_{j\in L(t)} f_a^{(j)}\Big)\,dt.$$

The bound involving $\bar\omega(p_a;\lambda)$ holds because of the following: If $p_a(f) = 0$ then it follows that

$$\int_0^x p_a(f+z)\,dz - \lambda\, p_a(x)\, x \leq 0,$$

since $\lambda \in \Lambda_2(\mathcal{C})$. Thus, the bound is true. If $f = 0$, then

$$\int_0^x p_a(z)\,dz - \lambda\, p_a(x)\, x \leq 0,$$

since $\lambda \geq 1$ and $p_a(\cdot)$ is nonnegative and nondecreasing. The case $p_a(f)\, f > 0$ follows from the definition.

Using the definition of $\delta(\mathcal{C})$ and applying $\delta(p_a)$ to the second term yields:

$$\leq \lambda\, \delta(\mathcal{C})\, C(\boldsymbol{x}) + \sum_{a\in A} \bar\omega(p_a;\lambda)\, \delta(p_a) \int_0^\infty \int_0^{\sum_{j\in L(t)} f_a^{(j)}} p_a(z)\,dz\,dt$$

$$\leq \lambda\, \delta(\mathcal{C})\, C(\boldsymbol{x}) + \delta(\mathcal{C})\, \bar\omega(\mathcal{C};\lambda)\, C(\boldsymbol{f}).$$

Applying this inequality to the optimal offline solution $\boldsymbol{x}$, rewriting, and taking the infimum over $\lambda \in \Lambda(\mathcal{C})$ yields the desired result for U-Seq. By Corollary 5.2 the result for U-Seq^2 follows as well. □

5.2. Polynomial Price Functions

To apply the result of Theorem 5.5, we need to bound $\bar\omega(\mathcal{C};\lambda)$. We start with price functions in $\mathcal{C}_1$, i.e., affine linear price functions.

Theorem 5.6. *For affine linear price functions in $\mathcal{C}_1$, the competitive ratio of* U-Seq *and* U-Seq^2 *are bounded from above by $3 + 2\sqrt{2}$.*

Proof. We are given price functions of the form $p_a(z) = c_1 z + c_0$, with $c_1 \geq 0$, and $c_0 \geq 0$. For bounding $\bar{\omega}(\mathcal{C}_1; \lambda)$ from above, we can assume $f, x > 0$. We will later determine $\lambda > 1$.

We evaluate $\bar{\omega}(\mathcal{C}_1; \lambda)$ for the constant term c_0 and the variable term $c_1 z$ separately, using a splitting argument as in Lemma 4.4. For the constant part, we have

$$\bar{\omega}(c_0; \lambda) = \sup_{f,x>0} \frac{(1-\lambda)\,x}{f}.$$

With the condition $\lambda > 1$, the above value is nonpositive. For the variable part $c_1 z$ we get

$$\bar{\omega}(c_1 z; \lambda) = \sup_{f,x>0} \frac{f\,x + \frac{1}{2}\,x^2 - \lambda\,x^2}{f^2}.$$

Defining $\mu := \frac{x}{f}$ (recall that $f > 0$), we obtain

$$\bar{\omega}(c_1 z; \lambda) = \max_{\mu \geq 0} \big(\mu + \tfrac{1}{2}\,\mu^2 - \lambda\,\mu^2\big).$$

For $\lambda > 1$ this is a strictly concave program with the unique solution $\mu = \frac{1}{2\lambda - 1}$. Inserting this solution yields

$$\bar{\omega}(c_1 z; \lambda) \leq \frac{1}{2\,(2\,\lambda - 1)}.$$

Thus, using Theorem 5.5 the competitive ratio is bounded from above by

$$\frac{2\,\lambda}{1 - \frac{1}{2\,\lambda - 1}} = \frac{2\,\lambda\,(2\,\lambda - 1)}{2\,\lambda - 2}.$$

Choosing $\lambda = 1 + \frac{1}{2}\sqrt{2}$ as the minimizer of this expression, the claim is proven for U-Seq. Corollary 5.2 yields the result for U-Seq^2. ☐

We need the following lemma for the proof of Theorem 5.8 below.

Lemma 5.7. *Let p_a be a continuous and nondecreasing price function satisfying $p_a(c \cdot z) \geq c \cdot p_a(z)$ for all $c \in [0, 1]$. Then we have $\bar{\omega}(p_a; \lambda) \leq \frac{1}{4\,(\lambda - 1)}$ for $\lambda > 1$.*

Proof. In the following, we can assume $p_a(f)\,f > 0$ and $x > 0$ for bounding $\bar{\omega}$, since otherwise the claim is trivially true. We first consider the case in which $f \geq x$. In this case we define $\mu = \frac{x}{f} \in (0, 1]$. We obtain

$$\begin{aligned}
\bar{\omega}(p_a; \lambda) &= \sup_{f>0,\ \mu\in(0,1]} \frac{\int_0^{\mu f} p_a(f+z)\,dz - \lambda\,p_a(\mu\,f)\,\mu\,f}{p_a(f)\,f} \\
&\leq \sup_{f>0,\ \mu\in(0,1]} \frac{\big(p_a((1+\mu)\,f) - \lambda\,p_a(\mu\,f)\big)\,\mu\,f}{p_a(f)\,f} \\
&\leq \sup_{f>0,\ \mu\in(0,1]} \frac{\big((1+\mu)\,p_a(f) - \lambda\,\mu\,p_a(f)\big)\,\mu\,f}{p_a(f)\,f} \\
&= \sup_{\mu\in(0,1]} \big(1 + \mu - \lambda\,\mu\big)\,\mu.
\end{aligned}$$

Here we have used the assumption on p_a, which also implies that $p_a(cz) \leq c \cdot p_a(z)$ for all $c \geq 1$. The above problem is a concave program with optimal value $\frac{1}{4\,(\lambda-1)}$.

For the case in which $f \leq x$, we define $\mu = \frac{f}{x} \in (0,1]$. Then, using similar arguments we arrive at

$$\bar{\omega}(p_a;\lambda) = \sup_{x>0,\ \mu\in(0,1]} \frac{\big(p_a((1+\mu)\,x) - \lambda\, p_a(x)\big)\,x}{p_a(\mu\,x)\,\mu\,x} \leq \sup_{\mu>0} \frac{1+\mu-\lambda}{\mu^2}.$$

This problem is again a concave program with optimal value $\frac{1}{4\,(\lambda-1)}$. □

Theorem 5.8. *For continuous, nondecreasing, and concave price functions, the competitive ratios of* U-Seq *and* U-Seq^2 *are at most* $4 + 2\sqrt{3}$.

Proof. The set of functions satisfying the assumptions of Lemma 5.7 contain concave functions. Hence, $\bar{\omega}(\mathcal{C};\lambda)$ is bounded from above by $1/(4\,(\lambda-1))$. Furthermore, it is easy to show that for concave functions $\delta(\mathcal{C}) \leq 2$. Hence, Theorem 5.5 implies that the competitive ratio is bounded from above by

$$\frac{4\,\lambda\,(\lambda-1)}{2\,\lambda-3}.$$

Choosing $\lambda = \frac{1}{2}(3+\sqrt{3})$ as the minimizer of this expression and using Corollary 5.2, the claim is proven for U-Seq and U-Seq^2. □

We now study the case, where the price functions are in $\mathcal{C}_d$. We state the following useful lemma (see, for instance, Farzad et al. [12]):

Lemma 5.9. *For nonnegative numbers x and f, a positive integer d, and $\beta \in (0,1)$, we have*

$$(f+x)^d \leq \beta^{1-d}\,f^d + (1-\beta)^{1-d}\,x^d.$$

We derive an upper bound on the competitive ratio of U-Seq by analyzing the value $\bar{\omega}(p_a;\lambda)$ for price functions in $\mathcal{C}_d$.

Theorem 5.10. *The competitive ratios of* U-Seq *and* U-Seq^2 *for price functions in $\mathcal{C}_d$, $d \geq 2$, are at most $O((1.77)^d\,d^{d+1})$.*

Proof. We first bound the value $\bar{\omega}(p_a;\lambda)$ for price functions $p_a \in \mathcal{C}_d$. Using similar arguments as in the proof of Lemma 4.4, it suffices to bound the value $\bar{\omega}(p_a;\lambda)$ for monomial price functions $c_d\,z^d$, $c_d \geq 0$ and for $x, f > 0$. We start with the definition of $\bar{\omega}$ and apply Lemma 5.9 for some $\beta \in (0,1)$ to be determined later.

$$\begin{aligned}
\bar{\omega}(c_d\,z^d;\lambda) &= \sup_{f,x>0} \frac{\int_0^x c_d\,(f+z)^d\,dz - \lambda\,c_d\,x^{d+1}}{c_d\,f^{d+1}} \\
&\leq \sup_{f,x>0} \frac{\int_0^x \beta^{1-d}\,f^d + (1-\beta)^{1-d}\,z^d\,dz - \lambda\,x^{d+1}}{f^{d+1}} \\
&\leq \sup_{f,x>0} \frac{\beta^{1-d}\,f^d\,x - (\lambda - \frac{1}{d+1}\,(1-\beta)^{1-d})\,x^{d+1}}{f^{d+1}}.
\end{aligned}$$

Defining $\mu := \frac{x}{f}$ (using $f > 0$), we obtain

$$\bar{\omega}(c_d\,z^d;\lambda) \leq \max_{0\leq\mu} \big(\beta^{1-d}\mu - (\lambda - \tfrac{1}{d+1}\,(1-\beta)^{1-d})\,\mu^{d+1}\big).$$

Calculating the maximum leads to:

$$\bar{\omega}(c_d\, z^d; \lambda) \leq \frac{\beta^{1-d}\, d}{d+1} \Big(\frac{\beta^{1-d}}{\big(\lambda - \frac{1}{d+1}\,(1-\beta)^{1-d}\big)\,(d+1)} \Big)^{\frac{1}{d}}.$$

We define $\lambda := \alpha^d\, d^d$ and $\beta := 1 - \frac{1}{\alpha\, d}$ for some constant $\alpha \geq 1$. Then, it follows that

$$\beta^{1-d} = \left(1 - \frac{1}{\alpha\, d}\right)^{1-d} \leq e^{\frac{1}{\alpha}}.$$

Thus, we have

$$\begin{aligned} \bar{\omega}(c_d\, z^d; \lambda) &\leq \frac{e^{\frac{1}{\alpha}}\, d}{d+1} \left(\frac{e^{\frac{1}{\alpha}}}{\big((\alpha\, d)^d - \frac{1}{d+1}\,(\alpha\, d)^{d-1}\big)\,(d+1)} \right)^{\frac{1}{d}} \\ &= \frac{e^{\frac{1}{\alpha}}\, d}{d+1} \left(\frac{e^{\frac{1}{\alpha}}}{(\alpha\, d)^d\,\big(1 - \frac{1}{(d+1)\,\alpha\, d}\big)\,(d+1)} \right)^{\frac{1}{d}}. \end{aligned}$$

Using $(1 - \frac{1}{(d+1)\,\alpha\, d})\,(d+1) \geq e^{\frac{1}{\alpha}}$ for $d \geq 2$ and $\alpha \geq \frac{3}{2}$, we get

$$\bar{\omega}(c_d\, z^d; \lambda) \leq e^{\frac{1}{\alpha}}\, \frac{1}{\alpha\,(d+1)}.$$

In order to apply Theorem 5.5 we have to show that $\lambda \in \Lambda(\mathcal{C}_d)$. First, $\lambda \geq 1$ implies $\lambda \in \Lambda_2(\mathcal{C}_d)$. Using $\delta(\mathcal{C}_d) \leq d+1$ (by Lemma 4.3), it follows that $\delta(\mathcal{C}_d)\, \bar{\omega}(\mathcal{C}_d; \lambda) \leq e^{\frac{1}{\alpha}}\, \frac{1}{\alpha}$. Thus, the set $\Lambda(\mathcal{C}_d)$ is contained in

$$\{\lambda = (\alpha\, d)^d \;:\; \alpha \geq \tfrac{3}{2},\; e^{\frac{1}{\alpha}}\, \tfrac{1}{\alpha} < 1\}.$$

Applying Theorem 5.5 yields:

$$C(\boldsymbol{f}) \leq \inf_{\{\alpha \geq \frac{3}{2} \,:\, e^{\frac{1}{\alpha}}\, \frac{1}{\alpha} < 1\}} \frac{(d+1)\,\alpha^d\, d^d}{1 - e^{\frac{1}{\alpha}}\, \frac{1}{\alpha}}\, C(\boldsymbol{x}).$$

If we use $\alpha = 1.77$, we get an upper bound of $O(1.77^d\, d^{d+1})\, C(\boldsymbol{x})$. □

For the unsplittable variant of the generalized load balancing problem defined in Section 4.4, we obtain the following result.

Corollary 5.11. *For the unsplittable online load balancing problem in general networks, time windows, and the ℓ_d-norm,* U-Seq *and* U-Seq^2 *are*

$$\big(1.77 - e^{\frac{1}{1.77}}\big)^{-\frac{1}{d+1}} \cdot 1.77 \cdot d = O(d)$$

competitive.

The proof is similar to the proof of Corollary 4.15. Note that this bound is the same as the one presented in Awerbuch et al. [4]. There are two differences, however: On the one hand, our bound holds for general network topologies. On the other hand, we only generalize the related machine case (due to flow conservation constraints) and do not cover the unrelated case.

5.3. Lower Bounds for the Unsplittable MRTW

In this section, we consider lower bounds for the the unsplittable MRTW.

Corollary 5.12. *The online algorithms* U-Seq *and* U-Seq^2 *are at least* $(d+1)^{d+1}$*-competitive for price functions in* $\mathcal{C}_d$*.*

Proof. Since one can approximate splittable flows by unsplittable flows with sufficiently small demands, the lower bounds of Theorem 4.18 carry over. □

Note that there is a gap between the upper bound of Theorem 5.10 and the above lower bound. For affine linear price functions, however, one can adapt a construction of Farzad et al. [12] to show that the upper bound of $3+2\sqrt{2}$ (see Theorem 5.6) is tight.

To obtain lower bounds for *any* deterministic online algorithm, we use the following result.

Theorem 5.13 (Awerbuch et al. [4])**.** *Any deterministic online algorithm for the unsplittable* OLB *under the* ℓ_d*-norm has competitive ratio of at least* $0.5307\,d$*.*

It can easily be seen that the relations derived in Section 4.4 remain true for the unsplittable variant of OLB, which yields the following result.

Corollary 5.14. *Every deterministic online algorithm for the unsplittable* MRTW *with price functions in* $\mathcal{C}_d$ *has a competitive ratio of at least*

$$(0.5307\,(d+1))^{d+1}.$$

Recently, Caragiannis [7] improved the lower bounds of Awerbuch et al. [4] for the more general case of online load balancing problem with *unrelated* machines. These lower bounds are equal to the upper bounds presented in [4, 7]. Caragiannis's proof, however, strongly relies on the property that jobs have different weights on different machines. This setting cannot be incorporated into our model. It is indeed likely that due to flow conservation constraints for multicommodity flows the two models are conceptually different in general.

6. Conclusions and Open Questions

Let us briefly discuss the tightness of the results obtained in this paper. It turns out that for the splittable MRTW, the analysis of Seq and Seq^2 is asymptotically (w.r.t. the number of commodities) tight (Theorem 4.2 and 4.18). For the unsplittable MRTW, the analysis for U-Seq and U-Seq^2 is only tight for $d \le 1$ (Theorem 5.10, 5.14, and 5.6). Furthermore, there is a gap between the lower bound on the competitive ratio for any deterministic online algorithm and the best upper bounds for both variants. It is an open question whether any of these bounds can be improved.

As mentioned earlier, the introduction of time windows generalizes the setting of Harks et al. [17, 18]. We now discuss the structural differences arising from this generalization. For polynomial price functions and general networks, we were not able to obtain larger lower bounds for the greedy online algorithms in the presence of time windows. For the parallel arc case, however, we proved that Seq does not compute an optimal solution anymore.

On the other hand, for polynomial price functions we could extend known upper bounds to this more general case. We conclude that the generalization to time windows makes a structural difference, but this difference is not well understood yet. A deeper investigation of this issue seems to be an interesting problem for the future.

Concerning the generalization to the case of routing more than one commodity per round, it turns out that neither SEQ nor SEQ^2 (U-SEQ nor U-SEQ^2) dominates the other (Corollary 3.5 and 5.2). Furthermore, the competitive ratios of SEQ and SEQ^2 (and U-SEQ/U-SEQ^2) are the same in the worst case (for the price functions studied in this paper). Thus, it seems that the power of SEQ is in general not increased by the possibility of routing several commodities simultaneously. There is, however, an algorithmic difference for the unsplittable MRTW. The subproblems that have to be solved by U-SEQ are NP-hard, while U-SEQ^2 is polynomially implementable and thus gives a polynomial time approximation algorithm.

In this paper, we have derived explicit upper bounds on the competitive ratio for the considered algorithms for price functions that are either polynomials with nonnegative coefficients or concave. This provides a rich class of functions. Results for other continuous and nondecreasing functions remain an open issue. For instance, although it seems unlikely, it is still possible that SEQ^2 has a better competitive ratio than SEQ for price functions different from the ones studied in this paper.

REFERENCES

[1] S. ALBERS, *Online algorithms: a survey*, Math. Program., 97 (2003), pp. 3–26.

[2] E. ALTMAN, T. BASAR, T. JIMENEZ, AND N. SHIMKIN, *Competitive routing in networks with polynomial costs*, IEEE Trans. Automat. Control, 47 (2002), pp. 92–96.

[3] A. AVIDOR, Y. AZAR, AND J. SGALL, *Ancient and new algorithms for load balancing in the L_p norm*, Algorithmica, 29 (2001), pp. 422–441.

[4] B. AWERBUCH, Y. AZAR, E. GROVE, M.-Y. KAO, P. KRISHNAN, AND J. VITTER, *Load balancing in the L_p norm*, in Proc. 36th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 1995, pp. 383–391.

[5] B. AWERBUCH, Y. AZAR, AND S. PLOTKIN, *Throughput-competitive on-line routing*, in Proc. 34th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 1993, pp. 32–40.

[6] Y. AZAR, A. EPSTEIN, AND L. EPSTEIN, *Load balancing of temporary tasks in the ℓ_p norm*, Theor. Comput. Sci., 361 (2006), pp. 314–328.

[7] I. CARAGIANNIS, *Better bounds for online load balancing on unrelated machines*, in Proc. of 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2008.

[8] I. CARAGIANNIS, M. FLAMMINI, C. KAKLAMANIS, P. KANELLOPOULOS, AND L. MOSCARDELLI, *Tight bounds for selfish and greedy load balancing*, in Proc. of the 33rd International Colloquium of Automata, Languages and Programming (ICALP), vol. 4051 of LNCS, Springer, Berlin, 2006, pp. 311–322.

[9] J. R. CORREA, A. S. SCHULZ, AND N. STIER MOSES, *On the inefficiency of equilibria in congestion games*, in Integer programming and combinatorial optimization. 11th international IPCO conference, Berlin, M. Jünger and V. Kaibel, eds., vol. 3509 of LNCS, Springer, Berlin, 2005, pp. 167–181.

[10] J. R. CORREA, A. S. SCHULZ, AND N. E. STIER MOSES, *Selfish routing in capacitated networks*, Math. Oper. Res., 29 (2004), pp. 961–976.

[11] S. DAFERMOS AND F. SPARROW, *The traffic assignment problem for a general network*, J. Res. Natl. Bur. Stand., Sect. B, 73 (1969), pp. 91–118.

[12] B. FARZAD, N. OLVER, AND A. VETTA, *Flow-free selfish routing*. Manuscript, 2006.

[13] B. Fortz and M. Thorup, *Optimizing OSPF/IS-IS weights in a changing world*, IEEE J. Sel. Area Comm., 20 (2002), pp. 756–767.

[14] ———, *Increasing internet capacity using local search*, Comput. Optim. Appl., 29 (2004), pp. 13–48.

[15] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer, Heidelberg, 2nd ed., 1993.

[16] T. Harks, *On the price of anarchy of network games with nonatomic and atomic players*, tech. rep., available at Optimization Online, 2007.

[17] T. Harks, S. Heinz, and M. E. Pfetsch, *Competitive online multicommodity routing*, in Proc. Workshop on Approximation and Online Algorithms (WAOA) 2006, T. Erlebach and C. Kaklamanis, eds., vol. 4368 of LNCS, Springer, Berlin Heidelberg, 2006, pp. 240–252.

[18] T. Harks, S. Heinz, and M. E. Pfetsch, *Competitive online multicommodity routing*, ZIB Report 07-16, Zuse Institute Berlin, 2007.

[19] T. Harks and L. A. Végh, *Nonadaptive selfish routing with online demands*, in Proc. of the Fourth Workshop on Combinatorial and Algorithmic Aspects of Networking (CAAN), vol. 4051 of LNCS, Springer, Berlin, 2007, pp. 27–45.

[20] B. Kalyanasundaram and K. Pruhs, *Speed is as powerful as clairvoyance*, J. ACM, 47 (2000), pp. 617–643.

[21] A. Karlin, M. Manasse, L. Rudolph, and D. D. Sleator, *Competitive snoopy caching*, Algorithmica, 3 (1988), pp. 79–119.

[22] N. Olver, *The price of anarchy and a priority-based model of routing*, Master's thesis, McGill Universty, Montréal, Canada, 2006.

[23] T. Roughgarden and E. Tardos, *How bad is selfish routing?*, J. ACM, 49 (2002), pp. 236–259.

[24] D. D. Sleator and R. E. Tarjan, *Amortized efficiency of list update and paging rules*, Commun. ACM, 28 (1985), pp. 202–208.

[25] S. Suri, C. Toth, and Y. Zhou, *Selfish load balancing and atomic congestion games*, Algorithmica, 47 (2007), pp. 79–96.

[26] A. Yahaya, T. Harks, and T. Suda, *iREX: Efficient inter-domain QoS automation using economics*, in Proc. of IEEE Globecom, 2006.

[27] A. Yahaya and T. Suda, *iREX: Inter-domain QoS automation using economics*, in Proc. IEEE Consumer Communications and Networking Conference, 2006, pp. 96–101.

(T. Harks) Institute of Mathematics, Technical University Berlin, Strasse des 17. Juni 136, 10623 Berlin, Germany.

E-mail address: harks@math.tu-berlin.de

(S. Heinz, M. E. Pfetsch) Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany.

E-mail address: {heinz,pfetsch}@zib.de

(T. Vredeveld) Maastricht University, Department of Quantitative Economics,, P.O. Box 616, 6200 MD Maastricht, The Netherlands.

E-mail address: t.vredeveld@ke.unimaas.nl