Oligopolistic Competitive Packet Routing

Britta Peis
Department of Management Science, RWTH Aachen
Kackertstraße 7, 52072 Aachen, Germany
peis@oms.rwth-aachen.de

Bjoern Tauer
Department of Management Science, RWTH Aachen
Kackertstraße 7, 52072 Aachen, Germany
bjoern.tauer@oms.rwth-aachen.de

Veerle Timmermans
Department of Management Science, RWTH Aachen
Kackertstraße 7, 52072 Aachen, Germany
veerle.timmermans@oms.rwth-aachen.de

Laura Vargas Koch
Department of Management Science, RWTH Aachen
Kackertstraße 7, 52072 Aachen, Germany
laura.vargas@oms.rwth-aachen.de

Abstract

Oligopolistic competitive packet routing games model situations in which traffic is routed in discrete units through a network over time. We study a game-theoretic variant of packet routing, where in contrast to classical packet routing, we are lacking a central authority to decide on an oblivious routing protocol. Instead, selfish acting decision makers ("players") control a certain amount of traffic each, which needs to be sent as fast as possible from a player-specific origin to a player-specific destination through a commonly used network. The network is represented by a directed graph, each edge of which being endowed with a transit time, as well as a capacity bounding the number of traffic units entering an edge simultaneously. Additionally, a priority policy on the set of players is publicly known with respect to which conflicts at intersections are resolved. We prove the existence of a pure Nash equilibrium and show that it can be constructed by sequentially computing an integral earliest arrival flow for each player. Moreover, we derive several tight bounds on the price of anarchy and the price of stability in single source games.

2012 ACM Subject Classification Networks → Network algorithms

Keywords and phrases Competitive Packet Routing, Nash Equilibrium, Oligopoly, Efficiency of Equilibria, Priority Policy

Digital Object Identifier 10.4230/OASIcs.ATMOS.2018.13

1 Introduction

One of the fundamental problems in parallel and distributed systems lies in the transport of discrete traffic units ("packets") through a network over time. From a centralized optimization perspective, the design of routing protocols requires two kinds of decisions: first, an origin-destination path needs to be selected for each of the packets, and second, priority policies need to be defined to resolve conflicts whenever more packets than the link-capacity allows are going to traverse the same link simultaneously.

18th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2018).
Editors: Ralf Borndörfer and Sabine Storandt; Article No. 13; pp. 13:1–13:22
OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

We model the network by a directed graph $G = (V, E)$ whose edges correspond to the links of the network. Each edge $e \in E$ is equipped with a certain bandwidth $u_e > 0$ denoting the maximal number of packets that are allowed to enter edge e simultaneously, and a certain transit time $\tau_e \geq 0$ denoting the time needed for a single packet to traverse e. Each packet must be sent through the network from its origin $s_i \in V$ to its destination $t_i \in V$ along a single path P_i selected from the collection $\mathcal{P}_i \subseteq 2^{|E|}$ of all simple s_i-t_i-paths. We assume that time is discrete and that all packets take their steps simultaneously. Thus, it suffices to consider integral capacities and travel times. The corresponding packet routing problem is to minimize the makespan of such a routing protocol, which is the latest point in time when a packet reaches its destination vertex. This problem is also known under the name *quickest integral multi-commodity flow over time* (see e.g. [4]).

When considering packet routing problems, like routing traffic in a road network, it is natural to view these problems from a game-theoretical perspective. In particular, as it might well be the case that there is no central authority which predescribes a routing protocol. Instead, packets are routed through the network by selfish acting decision makers ("players") each of which aiming at sending the packets under her control as fast as possible from the player-specific origin to the player-specific destination. Such situations can be modeled by *competitive packet routing games*, a special class of non-cooperative strategic games. In a competitive packet routing game, the network and the forwarding policy are publicly known. Each of the players $i \in N = \{1, \ldots, n\}$ decides on the routes along which the k_i packets under her control are to be routed from origin s_i to destination t_i.

Packet routing games usually restrict to the setting where each player controls exactly one packet. In this paper, we consider the more general setting where each player $i \in N$ controls an arbitrary integral amount of k_i packets which all need to be routed along paths in $\mathcal{P}_i$. We call these games *oligopolistic competitive packet routing games* to distinguish between our model and the model of competitive packet routing games investigated in [6]. The individual goal for each player is to minimize the average arrival time of the packets under her control, which corresponds to the computation of an earliest arrival schedule [9]. As a forwarding policy, we assume in our model that a global priority list π on the players is given according to which conflicts at intersections are resolved. That is, when more packets seek to enter an edge than the capacity allows for, packets belonging to players higher on the priority list go first. In these games, we study the drawbacks of the absence of a central authority, and the benefits of coordination between players. This analysis is motivated by future road traffic scenarios where instead of individual cars, private companies own fleets with a large number of autonomous vehicles. Similar to the development of the commercialization of the internet (and the possible abolition of net neutrality), one can think of a system where higher paying fleet owners gain benefits (priority) over non-paying fleet owners. As a city you are interested in the performance of such a prioritized system.

Contributions

A strategy where no player can unilaterally deviate to decrease her cost is called a *pure Nash equilibrium* (*equilibrium*, for short). In Section 3, we show that an equilibrium exists and that it can be constructed within pseudo-polynomial time by sequentially computing an earliest arrival flow for each player. In Section 4 and Section 5, we measure the efficiency of equilibria by comparing the best and worst total cost under an equilibrium state with the minimal total cost achievable by a central authority. The corresponding ratios are usually referred to as *Price of Stability (PoS)* [cf. [1]] and the *Price of Anarchy (PoA)* [cf. [11]], respectively.

In Section 4 we consider games in which all players share a common source and a common sink ("single commodity games"). We prove that the PoS in single commodity games is equal to 1, while the PoA is bounded from above by n. To show the tightness of the PoA, we provide an example in which the PoA converges to n with increasing number of packets. For the case where all players have identical demands, i.e., where $k_i = k_j$ for all $i, j \in N$, we prove that the PoA is bounded from above by $\frac{1}{2}(n+1)$ and give a matching lower bound example. Note that these bounds depend on the number of players, but are independent of the number of packets to be routed through the network. Thus, even for a very high number of packets we get a low price of anarchy if the number of players is small.

Lastly, in Section 5, we study games in which all players share a common source s, but might have player-specific sinks t_i ("single source games"). For single source games, we give an example where the PoS grows to 2 with increasing number of packets. The PoA might also be larger than for single commodity games. We even give an algorithm that computes, given the demands of all players, an example maximizing the PoA for the given set of demands.

Related Work

Packet routing has received a vast amount of attention in the past decades. A break-through result is due to Leighton, Maggs and Rao [13], who prove the existence of a routing protocol for fixed paths, whose makespan is a constant-factor approximation in terms of the natural lower bound $(C + D)/2$. Here, C denotes the *congestion*, i.e., the maximum number of packets traversing the same edge, and D denotes the *dilation*, i.e., the length of the longest path along which a packet is routed. This result has been improved and simplified several times in the past (see, e.g., [19, 16, 7, 17]). For the more general problem where paths are not fixed, Srinivasan and Teo [21] show that a constant factor approximation is still possible. To prove this result they use the fact that it is sufficient to find paths which minimize the sum of congestion and dilation. Koch et al. [10] extend this result to a more general setting, where messages that consist of several packets need to be routed through a network. In contrast to our model, they require that all packets of a message wait at the head of each traversed link until the last packet of the message arrived.

A game-theoretic perspective on packet routing can be found in the pioneering work of Hoefer et al. [8]. Here, they start with network congestion games (see Roughgarden [18] for an introduction) and generalize this model to a variant over time. More details on this development can be found in [8]. Similar to our competitive packet routing model, the model in [8] considers players $i \in N$, and each player is associated with an origin vertex s_i, a destination vertex t_i, and a player-specific weight w_i. However, in contrast to our model, the capacity on each link does not bound the number of packets allowed to traverse the link simultaneously at each integral time step. Rather, it bounds the total load on an edge induced by packets traversing this edge at each point in time. The authors analyze four different forwarding policies (FIFO, equal time sharing, (non-) preemptive global ranking), they focus on the existence of Nash equilibria and the convergence of best responses. Kulkarni et al. [12] extend the model of Höfer et al. and bound the price of anarchy, using LP duality. Lastly, Harks et al. [6] investigates the special class of competitive packet routing in which each player controls exactly one packet. They study existence, efficiency, and computability of equilibria with respect to both local (i.e. edge-dependent) and global priority lists on the players. For both forwarding policies, they analyze the existence of equilibria and establish bounds on the price of anarchy and the price of stability using the techniques introduced by Kulkarni et al [12]. A more detailed comparison can be found in the respective chapters.

2 Preliminaries

The Model

A *multi commodity oligopolistic competitive packet routing game* $\mathcal{G}$ is represented by the tuple: $\mathcal{G} := (G, N, (s_i, t_i, k_i)_{i \in N}, \pi)$, where $G := (V, E, (\tau_e)_{e \in E}, (u_e)_{e \in E})$ is a directed graph that consists of a set of nodes V and edges E, where each edge $e \in E$ is endowed with an integral transit time $\tau_e \geq 0$ and an integral capacity $u_e > 0$. The transit time of an edge denotes the time it takes for each player to traverse this edge. The capacity is a limit on the number of packets that can enter an edge at each integral time step. We use N to denote the set of players, where each player $i \in N$ has a player-specific source and sink $s_i, t_i \in V$. Additionally, each player has a set of k_i identical packets she desires to send from s_i to t_i. We denote this set by K_i. Lastly, as a forwarding policy, we are given a priority list $\pi \in \Pi_n$, where Π_n is the set of all different orderings on n players. Whenever more packets desire to enter an edge than the capacity allows for, packets of players higher in the priority list can go first. Without loss of generality, we assume that players are numbered according to their position in the priority list π.

A feasible strategy x_i of a player $i \in N$ determines for every packet in K_i a simple s_i-t_i-path, together with a release time, i.e., the time the packet should start trying to traverse its assigned path. That is, player i decides on a path vector $P_i \in \mathcal{P}_i^{k_i}$, where $\mathcal{P}_i$ denotes the set of all simple s_i-t_i-paths. Additionally, player i decides on a release time for every packet by selecting a release-time vector $R_i \in \mathbb{N}_{\geq 0}^{k_i}$. Thus, the set of feasible strategies of player i can be described as $\mathcal{S}_i(k_i) := \left\{ x_i = (P_i, R_i) \mid P_i \in \mathcal{P}_i^{k_i}, R_i \in \mathbb{N}_{\geq 0}^{k_i} \right\}$.

The combined strategy space is denoted by $\mathcal{S} := \prod_{i \in N} \mathcal{S}_i(k_i)$ and additionally we denote by $x := (x_i)_{i \in N}$ the overall strategy profile. In a strategy profile x, each packet $\ell \in K_i$ travels over its assigned path to its destination. We let $C_{i,\ell}(x)$ denote its arrival time at sink t_i. The goal of each player is to minimize the sum of the arrival times of her packets $C_i(x) := \sum_{\ell \in K_i} C_{i,\ell}(x)$. The *social cost* of strategy profile $x \in \mathcal{S}$ is $C(x) = \sum_{i \in N} C_i(x)$, i.e., the total cost of all players. A profile $x \in \mathcal{S}$ minimizing the social cost is called *social optimum*.

Note that the arrival time of each packet is uniquely determined by embedding the strategies of the players in graph G. We embed the players one by one in order of their priority list and for every player we embed the packets in order of the strategy vector (assuming a decreasing priority) starting at their respective release time. In our model, packets are not allowed to wait at any intermediate node unless necessary. Thus, such an embedding is unique.

As usual in game theory, for every $i \in N$, we write $\mathcal{S}_{-i}(k_{-i}) := \prod_{j \neq i} \mathcal{S}_j(k_j)$ and $x = (x_i, x_{-i})$ meaning that $x_i \in \mathcal{S}_i(k_i)$ and $x_{-i} \in \mathcal{S}_{-i}(k_{-i})$. A strategy profile x is called a *Nash equilibrium* whenever no player can unilaterally deviate and decrease her own cost, i.e. $C_i(x_i, x_{-i}) \leq C_i(y_i, x_{-i})$ for all $y_i \in \mathcal{S}_i(k_i)$. A pair $(x, (y_i, x_{-i}))$ is called an *improving move* when $C_i(y_i, x_{-i}) < C_i(x)$. A strategy x_i of player i is called a *best response* to x_{-i} whenever $x_i \in \arg\min_{y_i \in \mathcal{S}_i(k_i)} \{C_i(y_i, x_{-i})\}$. Thus, a profile x is a Nash equilibrium if and only if there is no player that has an improving move, or equivalently, if each player plays a best response.

▶ **Example 1.** Consider the single commodity game $\mathcal{G}$ on directed graph G depicted in Figure 1, where the transit times are depicted in the picture, and the capacity of each edge is equal to one. We consider two players, each controlling exactly one packet that needs to be routed from the common source s to the common sink t. As stated before, we assume the

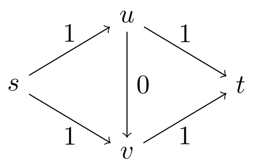

Figure 1 Graph G.

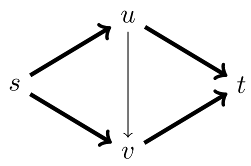

Figure 2 Social optimum.

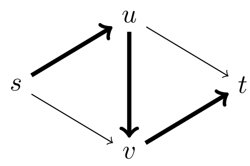

Figure 3 Strategy in a NE.

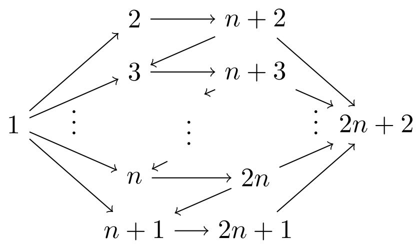

Figure 4 Graph $BG(n)$.

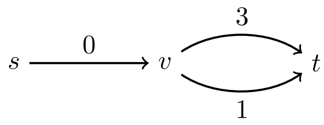

Figure 5 Release times as part of the strategy.

players to be numbered according to their spot in the priority list, hence, player 1 has priority over player 2. Note that the first player has three optimal strategies: she selects release time zero and either uses one of the parallel paths (Figure 2) or the path that intersects with both of these paths (Figure 3). If she chooses one of the parallel paths, the second player can take the other parallel path, resulting in a socially optimal equilibrium $x \in \mathcal{S}$ with $C_1(x) = C_2(x) = 2$. If she uses the the zig-zag path depicted in Figure 3 instead, she harms player 2 who cannot arrive at t before time step 3. If, for example, the second player selects release time zero, and travels along either path, we result in an equilibrium x' with social cost $C_1(x') + C_2(x') = 2 + 3 = 5$. Thus, $PoS = 1$ and $PoA \geq \frac{5}{4}$.

The graph G depicted in Figure 1 is a well-known graph, famous from the Braess-paradox, and is used several times to prove lower bounds on the price of anarchy, e.g. [12]. In the rest of this paper we use this graph, and an extension of it several times. Therefore, we define $BG(n)$ as a graph on $2n+2$ vertices with four types of edges $E_{BG(n)} = E_1(n) \cup E_2(n) \cup E_3(n) \cup E_4(n)$, where: $E_1(n) := \{(1, v) \mid v \in \{2, \dots, n+1\}$, $E_2(n) := \{(v, v+n) \mid v \in \{2, \dots, n+1\}$, $E_3(n) := \{(v, v-n+1) \mid v \in \{n+2, \dots, 2n\}$, $E_4(n) := \{(v, 2n+2) \mid v \in \{n+2, \dots, 2n+1\}$. Note that graph $BG(n)$ has n parallel paths from node 1 to $2n+2$, and one path that intersects all n parallel paths. A visualisation of graph $BG(n)$ can be found in Figure 4.

As stated in the model, players do not only choose a path in the network, but also a release time for each packet, i.e., the time at which a packet starts traversing its assigned path. This brings no advantage regarding the cost function of a player. Though, by allowing players to set a release time for each packet, friendly players have the option not to congest the network unnecessarily. Moreover, players might prefer to wait at the source instead of waiting at intermediate nodes. As is proven in Section 3, it also allows us to compute social optima in all single commodity games. We give an example that illustrates the use of setting release times for packets: Consider the graph depicted in Figure 5, with four players owning one packet each. The first edge has capacity two and the other edges have capacity one. The transit times of the edges are depicted in the network. We denote the path taking the lower edge (v, t) as p_1 and the path taking the upper edge as p_2. A possible equilibrium is $x_1 = (p_1, 0)$, $x_2 = (p_1, 1)$, $x_3 = (p_1, 2)$ and $x_4 = (p_2, 0)$ realizing arrival times $C_1 = 1$, $C_2 = 2$, $C_3 = 3$ and $C_4 = 3$. If players cannot choose release times, there is a unique equilibrium in which all players choose p_1, realizing arrival times $C_1 = 1$, $C_2 = 2$, $C_3 = 3$ and $C_4 = 4$.

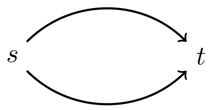

Figure 6 No PNE without global priority list.

Table 1 Three possible equilibria.

	Strategy 1	Strategy 2	Strategy 3
$(1,1)$	(top,0)	(top,0)	(top,0)
$(1,2)$	(top,0)	(bottom,0)	(bottom,0)
$(2,1)$	(bottom,0)	(top,0)	(bottom,0)
$(2,2)$	(bottom,0)	(bottom,0)	(top,0)

In Section 3 we prove that, whenever we are given a priority list on the players, Nash equilibria exist. In contrast, if the priority list is given on the set of packets instead of players, the existence of equilibria cannot be guaranteed.

▶ **Example 2.** Consider an oligopolistic packet routing game on the graph depicted in Figure 6. This graph has two edges: *top* and *bottom*, with both capacity and transit time equal to one. We assume there are two players that both want to route two packets from source s to sink t. Let (i, ℓ) denote packet ℓ of player i. Assume that the priority over the packets is $\pi = ((1,1),(2,1),(1,2),(2,2))$. Note that in this network no player can decrease her costs by increasing the release time from zero. Further, note that in any equilibrium each edge is used by exactly two packets. This implies that without loss of generality there are three candidates for an equilibrium, which are depicted in Table 1.

Strategy 1 is not an equilibrium, as player 1 is better of by switching packet one to the other edge. Strategy 2 is not an equilibrium, as player 2 is better of by interchanging packet one and two. Strategy 3 is also not an equilibrium, as player 1 would be better of by switching packet one and two around. Hence, this game does not have a Nash equilibrium.

Due to the simplicity of the example, it is sensible to restrict our research to priority lists on players instead of packets.

Flows over time and earliest arrival flows

In an oligopolistic packet routing game, a player sends a set of k_i packets from a source s_i to a sink t_i. Thus, every feasible strategy is an integral s_i-t_i-flow over time of flow value k_i. We shortly introduce flows over time, also known under the name *dynamic flows.* For a more detailed introduction on static and dynamic flows, we refer to Skutella [20].

▶ **Definition 3.** Given a graph $G = (V, E, (\tau_e)_{e \in E}, (u_e)_{e \in E})$ with transit times and capacities, an *integral s-t-flow over time* is a set of functions $f_e : \mathbb{N}_{\geq 0} \to \mathbb{N}_{\geq 0}$ for all $e \in E$ satisfying the following two constraints:

$$f_e(\theta) \leq u_e \qquad \forall e \in E, \theta \in \mathbb{N}_{\geq 0}, \tag{1}$$

$$\sum_{e \in \delta^-(v)} \sum_{\theta=0}^{\xi-\tau_e} f_e(\theta) \geq \sum_{e \in \delta^+(v)} \sum_{\theta=0}^{\xi} f_e(\theta) \qquad \forall \xi \in \mathbb{Z}_{\geq 0}, v \in V \setminus \{s, t\}. \tag{2}$$

Here $\delta^+(v) := \{(v,u) \in E \mid u \in V\}$ and $\delta^-(v) := \{(u,v) \in E \mid u \in V\}$. The first inequality imposes the capacity constraint of the edges on the flow, and the second constraint represents the flow conservation property. If Equation (2) is fulfilled with equality, we say that strong flow conservation holds, implying that there is no waiting at intermediate nodes.

A special variant of flows over time are *earliest arrival flows.* Such an earliest arrival flow (EAF) maximizes the flow value arriving at the sink at each integral time step $\theta \in \mathbb{N}_{\geq 0}$. To be more precise, we define $A(f,T)$ to be the amount of flow that arrives at t on or before time T, e.g. $A(f,T) := \sum_{\theta=0}^{T} a(f,\theta)$, where $a(f,\theta)$ denotes the amount of flow arriving at the sink at time θ. We say that a feasible integral s-t-flow over time f fulfills the *earliest arrival property* whenever $A(f,\theta) \geq A(f',\theta)$ for all feasible integral s-t-flows f' and all $\theta \in \mathbb{N}_{\geq 0}$. An integral s-t-flow over time that satisfies strong flow conservation and fulfills the earliest arrival property is called an *integral earliest arrival s-t-flow* (s-t-EAF). Integral earliest arrival flows are guaranteed to exist in a single commodity network [5]. In such networks, an earliest arrival flow can be computed by Wilkinson's algorithm [23] when the capacities do not vary over time, and Tjandra's algorithm when capacities do vary over time [22]. In a multiple source, single sink setting, earliest arrival flows also exist, and can be computed when capacities do not change over time [14, 15]. In a multi commodity setting there are networks such that no earliest arrival flow exists [3].

3 Existence of Nash equilibria

Whenever each player has exactly one packet, Harks et al. [6] show that a pure Nash equilibrium exists and can be found using a sequence of shortest path computations. We prove the existence of pure Nash equilibria in multi commodity oligopolistic competitive packet routing games by exploiting the connection to earliest arrival flows. We start by showing how to compute a best response for player i by computing an s_i-t_i-EAF in a network with time-varying capacities. Afterwards, we prove that a pure Nash equilibrium can be obtained by sequentially computing such an earliest arrival flow for each player, in order of the priority list. Lastly, we show that in a single commodity game an earliest arrival flow minimizes the social cost function.

▶ **Theorem 4.** *In a multi commodity oligopolistic competitive packet routing game, a best response of a player $i \in N$ corresponds to an s_i-t_i-earliest arrival flow with time-varying capacities, and vice versa.*

Proof. Fix a player $i \in N$ and strategies $x_1, \dots, x_{i-1}$ of players higher in the priority list, arbitrarily. As mentioned before, we assume that players are ordered according to the priority list. Thus, players $j \in \{i+1, \dots, n\}$ cannot influence the travel time of packets controlled by player i. A best response of player i towards x_{-i} is therefore a strategy choice (or flow) x_i minimizing $\sum_{\ell \in K_i} C_{i,\ell}(x)$, i.e., the sum of arrival times of all packets in K_i. Obviously, this corresponds to minimizing the average arrival time $\frac{1}{k_i}\sum_{\ell \in K_i} C_{i,\ell}(x)$. In [9], it was shown that minimizing $\frac{1}{k_i}\sum_{\ell \in K_i} C_{i,\ell}(x)$ is equivalent to maximizing $\sum_{\theta \in \mathbb{N}_{\geq 0}} A_i(x,\theta)$, where $A_i(x,\theta) := |\{\ell \in K_i \mid C_{i,\ell}(x) \leq \theta\}|$ denotes the number of packets of player i arriving at sink t_i before time θ under strategy x. For sake of completeness, we present a proof for this fact in Appendix A.

It is well-known, that every s-t-network admits an s-t-earliest arrival flow even for the case of time-dependent capacities (cf. Tjandra [22]). By definition, an earliest arrival flow (EAF) is a flow maximizing $A_i(x,\theta)$ for every time θ. Thus, such an EAF maximizes the sum $\sum_{\theta \in \mathbb{N}_{\geq 0}} A_i(x,\theta)$ as well, and therefore corresponds to a best response of player i.

On the contrary, there can be no feasible flow other than an earliest arrival flow maximizing this sum, since the earliest arrival flow maximizes every single summand. As a consequence, every best response corresponds to an earliest arrival flow, and vice versa.

We can compute a best response x_i as follows: We embed the strategies $x_1, \dots x_{i-1}$ of players $\{1, \dots, i-1\}$ one by one in the network, in order of the priority list. Using the algorithm of Tjandra [22], we compute an s_i-t_i-EAF f in the resulting network with varying capacities. We decompose flow f in k_i paths $(p_\ell)_{\ell \in K_i}$, where w.l.o.g. we assume that the paths are ordered according to non-decreasing path lengths. Each packet $l \in K_i$ is assigned the release time r_ℓ according to its release time in the path decomposition of the earliest arrival flow. To show that the strategy $x_i = (p_\ell, r_\ell)_{\ell \in K_i}$ is a feasible one, we prove that packets never wait at intermediate nodes, unless the capacity of this edge is reduced due to a preceding player, and that all paths are cycle-free. For a proof of the cycle-freeness, we refer to the Appendix B. If all packets start according to their release dates, no packet of player i has to wait for another packet of player i, since f is an earliest arrival flow. Particularly, all packets take the same path and arrive at every intermediate node at the same point in time as their correspondent in the earliest arrival flow. So, the arrival pattern of the flow corresponding to x_i has the earliest arrival property and thus x_i is a best response for player i. The priority rules in the model are obeyed since the players are embedded one by one in order to the priority list. If a packet of a player i needs to wait due to a reduced capacity, this corresponds to a packet of a player $j < i$ using the edge. ◀

Thus, in order to compute a pure Nash equilibrium, we subsequently compute earliest arrival flows for the players in the order of the priority lists according to Theorem 4.

▶ **Corollary 5.** *Each multi commodity oligopolistic competitive packet routing game admits a pure Nash equilibrium. Moreover, a pure Nash equilibrium can be computed by calculating subsequently an earliest arrival flow for each player in the order of the priority list by using the algorithm of Tjandra [22]. The running time is within* $O(|E| \cdot |V| \cdot \sum_{i \in N} (S'_i + k_i)^2 \cdot k_i)$, *where* S'_i *is the length of a shortest* s_i-t_i*-path in the underlying network with capacities adapted according to the best responses of players in* $\{1, \dots, i-1\}$.

In order to achieve a social optimum, we assume there is one central authority who coordinates all packets. Note that this central authority still needs to take the priority rules into account. In single commodity games, we are able to compute a social optimum.

▶ **Theorem 6.** *In single commodity oligopolistic competitive packet routing games, a social optimum can be computed within pseudo-polynomial time.*

Proof. First, we assume that there is a central authority controlling all $K = \sum_{i \in N} k_i$ packets. According to Theorem 4, a strategy minimizing the social cost function for one player corresponds to an earliest arrival flow. As capacities are constant over time, we can compute an earliest arrival flow f by using Wilkinson's algorithm [23]. Note that this algorithm computes K shortest paths, and thus runs in pseudo-polynomial time. It is left to decompose this strategy into player specific strategies and check if the player specific strategies obey the priority rules. In order to do so, we find a path decomposition of flow f with a corresponding release time for each packet: $(p_q, r_q)_{1 \le q \le K}$, where the tuples are numbered according to the time the corresponding packet arrives at the sink. We define $F_i := \sum_{q=1}^{i-1} k_q$ and $x_i = (p_q, r_q)_{F_i < q \le F_i + k_i}$.

By the choice of the release times, we guarantee that a packet following the corresponding successive shortest path can traverse the network without being delayed by other packets. Hence, x realizes the arrival pattern of an earliest arrival flow while obeying the priority rules. The paths are cycle-free due to Appendix B. ◀

4 Efficiency in single commodity games

We discuss the price of stability (PoS) and the price of anarchy (PoA) in single commodity games in this section, and in single source multiple sink games in the subsequent section. First of all, similar as in the model of Harks et al. [6], it can easily be derived from Theorem 6 that the price of stability in single commodity games is equal to one.

▶ **Corollary 7.** *Each single commodity oligopolistic competitive packet routing game admits a socially optimal pure Nash equilibrium which can be computed via one earliest arrival flow computation.*

Proof. We compute a social optimum as described in the proof of Theorem 6, and prove that the resulting strategies form a Nash equilibrium. Observe that strategy x_i is a best response for player i, as she cannot decrease the arrival times of her packets due to the earliest arrival property of the total flow. Thus, strategy $(x_i)_{i \in N}$ is an equilibrium minimizing the social cost. ◀

We show that in the single commodity setting, the price of anarchy is bounded by n. Furthermore, we introduce an example such that, when the number of packets grows large, the price of anarchy in our example converges to n. We start by proving an upper bound on the price of anarchy. The proof is based on the following insight.

▶ **Lemma 8.** *Let NE be a Nash equilibrium for game $\mathcal{G}$ and let OPT be a socially optimal strategy profile constructed as described in the proof of Theorem 6. Then, for every player i and every packet $\ell \in K_i$, it holds that: $C_{i,\ell}(NE) \leq i \cdot C_{i,\ell}(OPT)$.*

A proof of Lemma 8 can be found in the Appendix C. Using Lemma 8 we prove an upper bound on the price of anarchy in single commodity oligopolistic competitive packet routing games.

▶ **Theorem 9.** *In single commodity oligopolistic competitive packet routing games, the price of anarchy is bounded from above by n.*

Proof. Let NE be the Nash equilibrium for single commodity game $\mathcal{G}$ that maximizes the social cost, and let OPT be a strategy profile that minimizes the social cost function. We prove that $\frac{C(NE)}{C(OPT)} \leq n$. We use Lemma 8 to obtain:

$$C(NE) \leq \sum_{i=1}^{n} \sum_{\ell \in K_i} i \cdot C_{i,\ell}(OPT) \leq n \cdot \sum_{i=1}^{n} \sum_{\ell \in K_i} C_{i,\ell}(OPT) = n \cdot C(OPT).$$

Thus, the price of anarchy has an upper bound of n. ◀

In Theorem 10 we state an example of a single commodity game where, if the total number of packets in the game grows large, the price of anarchy converges to n.

▶ **Example 10.** Consider a game with n players, where the first $n-1$ players have only one packet, and player n has k_n packets. All players need to route their packets from s to t in the Braess graph $BG(n + k_n - 1)$ depicted in Figure 7.

In an optimal solution, all packets traverse the $k_n + n - 1$ available parallel paths as depicted in Figure 8, incurring a social cost of $k_n + n - 1$. Note that it is also a viable option for the first $n - 1$ players to traverse the path as depicted in Figure 9. The k_n packets of

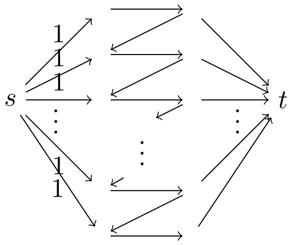

Figure 7 Graph BG.

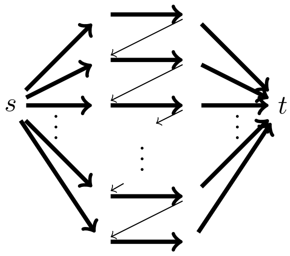

Figure 8 OPT.

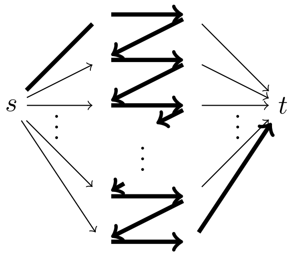

Figure 9 NE.

player n cannot arrive before time n, incurring a social cost of $\frac{1}{2}n(n-1)+nk_n$. Hence, in this example the price of anarchy is:

$$PoA = \frac{\frac{1}{2}n(n-1)+nk_n}{k_n+n-1} = n - \frac{\frac{1}{2}n(n-1)}{k_n+n-1}.$$

Note that when k_n grows large, this ratio converges to n. Also observe that this result generalizes the bound in [6], where all players own only one packet each ($k=1$).

Hence, the bound we prove in Theorem 9 is tight. In this example, we exploit the fact that the last player has far more packets than the others. Hence, it is reasonable to consider the special case that all players have the same number of packets k, i.e. $k_i = k_j$ for all $i, j \in N$. We denote such a game as a symmetric game, since the strategy spaces of all the players are identical. In a symmetric game the price of anarchy decreases to $\frac{1}{2}(n+1)$. This is an extension of the result of Harks et al. [6], which would give a price of anarchy of $\frac{1}{2}(kn+1)$ and coincide for $k=1$. In order to prove this statement, we first show that $\frac{1}{2}(n+1)$ is an upper bound on the price of anarchy.

▶ **Theorem 11.** *In symmetric oligopolistic competitive packet routing games the price of anarchy is bounded from above by* $\frac{1}{2}(n+1)$.

Proof. Let S be the length of the shortest s-t-path in the network. Assume that in an optimal strategy the first player has a_p packets arriving at time $S+p-1$, where p ranges from one up to some $q_1 \in \mathbb{N}_{>0}$, where $q_1 = \arg\min_{p\in\mathbb{N}_{>0}}\{a_{q'} = 0, \forall q' > p\}$. Thus $\sum_{p=1}^{q_1} a_p = k$, and within a time span of q_1 all packets of player 1 reach the sink. We say that q_1 is the *arrival spread* of player 1. Note that $1 \leq a_1 \leq a_2 \leq \cdots \leq a_{q_1-1}$, as at least one packet of player 1 can arrive at time S by taking the shortest path. Further, if a_p packets arrive at time $S+p-1$ at least so many packets can arrive at $S+p$ by choosing the same paths as the packets arriving at $S+p-1$ as long as there are enough packets left to fill up all paths.

As arrival times are increasing, the arrival times for all packets of the remaining $n-1$ players are at least $S+q_1-1$. Hence, we can find the following lower bound on the social cost: $C(OPT) \geq knS + \sum_{p=1}^{q_1} a_p(p-1) + (q_1-1)k(n-1)$.

In order to find an upper bound on the worst Nash equilibrium, we give an upper bound on the arrival times of the i'th player, in terms of the arrival times of the first player. We show that player i can always copy the strategy of player 1, but increase the release times by $(i-1)q_1$ time units. In general, we prove the following statement: if the first player has a_p packets arriving at $S+p-1$ as described above, then the i'th player can play the same strategy $(i-1)q_1$ time units later, with a_p packets arriving at $S+p-1+(i-1)q_1$.

We prove this statement by induction. In order to do so, we use the even stronger statement which says: player i can copy the strategy of the first player $(i-1)q_1$ time units later, without being delayed by any other player. Assume the induction hypothesis holds for the first $i-1$ players. Then we show that the i'th player can play the strategy of the first player, where the release times are increased by $(i-1)q_1$. Note that the arrival times of the first $i-1$ players are all strictly smaller than $S+(i-1)q_1$. On the contrary, we assume that there exists a packet ℓ of player i that needs to wait for a packet ℓ' by a previous player. This implies that, if packet ℓ' would not block packet ℓ, then packet ℓ could arrive on the original arrival time of packet ℓ', which is smaller than $S+(i-1)q_1$. As packet ℓ can only depart from s at release time $(i-1)q_1$, this would imply that the shortest s-t-path has a length smaller than S, which contradicts the fact that S is the length of shortest path in the network. Hence, player i can repeat the strategy of the first player $(i-1)q_1$ time units later without being delayed and thus with a_p packets arriving at $S+p-1+(i-1)q_1$. Furthermore, we know by Theorem 4 that a best response is equivalent to an earliest arrival flow. This guarantees that no packet of player i arrives later than $S+q_1-1+(i-1)q_1$. This gives us an upper bound on the total cost of any Nash equilibrium: $C(NE) \leq \sum_{i=1}^{n}\sum_{p=1}^{q_1} a_p(S+p-1+(i-1)q_1)$.

As we have a lower bound on the social cost of an optimal solution, and an upper bound of the cost in any Nash equilibrium, we can find an upper bound on the price of anarchy.

$$PoA \quad \leq \quad \frac{\sum_{i=1}^{n}\sum_{p=1}^{q_1} a_p(S+p-1+(i-1)q_1)}{knS+\sum_{p=1}^{q_1} a_p(p-1)+(q_1-1)k(n-1)}.$$

This fraction is maximized whenever $S=1$ and $q_1=1$, therefore the price of anarchy is bounded from above by $\frac{1}{2}(n+1)$. The technical argument for this claim can be found in the Appendix D. ◀

To observe that this result is tight, consider Braess graph $BG(n)$ (see Figure 7), where each edge has a capacity k. Assume that the edges leaving s have cost S. Since there are n disjoint paths with capacity k, all packets of all players reach the sink simultaneously at time S in a social optimal profile OPT (see Figure 8), resulting in a social cost $C(OPT) = nkS$. However, there is a profile in which all packets of each player take the path depicted in Figure 9, which turns out to be a Nash equilibrium NE. Here, the arrival time is $C_{i,\ell}(NE) = S+i-1$ for all players $i \in N$ and for all packets $\ell \in K_i$. Therefore $C(NE) = nk(S-1)+k\sum_{i=1}^{n} i$. If we choose $S=1$ we get the tight upper bound $PoA = \frac{1}{2}(n+1)$. Observe that this result generalizes the bound in [6], where all players own only one packet each $(k=1)$.

5 Efficiency in single source games

In this section, we consider games where players have a common source s, but player specific sinks t_i, $i \in N$. In general, earliest arrival flows do not necessarily exist in multi-commodity games, even if all commodities share a common source (see, e.g., [2]).

▶ **Example 12.** Consider the graph depicted in Figure 10 with unit capacities and travel times as shown in the picture. Assume one traffic unit needs to be send from s to t_1, and one unit from s to t_2. In order to maximize the amount of flow arriving after two units of time, we send one unit along edge (s,t_1) and one unit along (s,v,t_2) so that both units reach the respective sink after two time units. This flow does obviously not maximize the amount of flow reaching sink t_1 at time step $\theta = 1$. Thus, in this graph, no earliest arrival flow exists.

We extend this example to a single source competitive packet routing game on n players, and show that, in contrast to single commodity games, single source games do not necessarily admit socially optimal pure Nash equilibria.

▶ **Theorem 13.** *In oligopolistic competitive packet routing games with a global source s and player specific sinks $t_1,\ldots,t_n$, the price of stability is bounded from below by 2.*

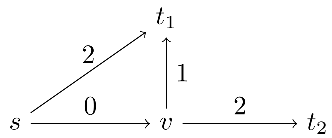

Figure 10 Network without an EAF.

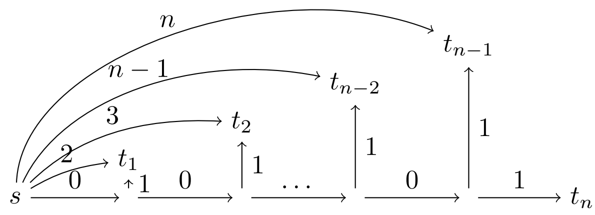

Figure 11 Network with a PoS converging to 2.

Proof. We consider the graph depicted in Figure 11, where the capacity of each edge is equal to one. We assume there are n players, where the first $n-1$ players control one packet, and the last player controls n packets. Note that each of the first $n-1$ players has two feasible strategies. Either she takes her direct s-t_i-route, or the path using the zero-length edges. The last player has only one feasible strategy.

In the optimal solution OPT, the first $n-1$ players all take the direct s-t_i-route, incurring a total cost of $n(n+1)-1$ for all players. In the unique Nash equilibrium NE, all players use their indirect route, incurring a total cost of $n(2n-1)$. Hence, the price of stability is:

$$PoS = \frac{C(NE)}{C(OPT)} = \frac{n(2n-1)}{n(n+1)-1} \geq \frac{2n-1}{n+1} = 2 - \frac{3}{n+1}.$$

Note that the last term converges to 2 when the number of players grows to infinity. ◀

In the remaining part of this section we focus on the price of anarchy. We show that, in contrast to single commodity games, the bound for the setting with equal demands coincides with the general bound. The tight bound turns out to be a fraction that depends on the number of packets k_i of player i and the number of players n. We present an algorithm that constructs a matching lower bound example for every given n and $(k_i)_{i\in N}$.

Similar as in the previous section, we define S_i to be the length of a shortest s-t_i-path in (G,τ), i.e., $S_i := \min_{P\in\mathcal{P}_i}\sum_{e\in P}\tau_e$. For each player $i \in N$, let OPT_i be an optimal strategy under the assumption that no other player exists, i.e., OPT_i is an integral earliest arrival flow with source s and sink t_i. Clearly, under flow OPT_i, at least one packet reaches the sink at time S_i. We are interested in the *arrival spread* q_i of flow OPT_i which is the length of the time interval in which packets are arriving at the sink under flow OPT_i. Here, q_1 corresponds to the arrival spread q_1 we defined in the proof of Theorem 11. To be more precise, we let $M_i(OPT_i)$ denote the makespan of player i in OPT_i, i.e., the latest point in time when a packet of player i reaches the sink. Then, we define $q_i := M_i(OPT_i) - S_i + 1$ to be the *arrival spread* of flow OPT_i. Let $C_{i,l}(OPT_i)$ denote the arrival time of the packet $l \in K_i$ under flow OPT_i.

▶ **Theorem 14.** *Let NE be an arbitrary pure Nash equilibrium in a single source multiple sink oligopolistic competitive packet routing game $\mathcal{G}$. Then, if $C_{i,\ell}(NE)$ denotes the arrival time of packet $l \in K_i$ under NE,*

$$C_{i,\ell}(NE) \leq C_{i,\ell}(OPT_i) + \sum_{k=1}^{i-1} q_k,$$

Proof. Recall that, under an equilibrium, each player $i \in N$ plays a best response towards the strategy choices of the players $j \in \{1,\ldots,i-1\}$ higher in the priority list. We prove this

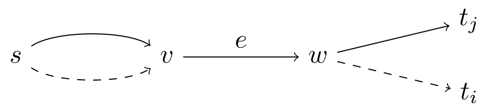

Figure 12 Two player with origin s interact at a common edge e in the network.

theorem by induction. For the first player, the statement trivially holds, since a best response corresponds to an s-t_1-EAF, so $C_{1,l}(NE) = C_{1,l}(OPT_1)$ for each packet $l \in K_1$ controlled by the first player. Assume that $C_{j,\ell}(NE) \leq C_{l,\ell}(OPT_j) + \sum_{k=1}^{j-1} q_k$ holds for each packet $l \in K_j$ controlled by a player $j \in \{1, \ldots, i-1\}$. To show that $C_{i,\ell}(NE) \leq C_{i,\ell}(OPT_i) + \sum_{k=1}^{i-1} q_k$ is true, it suffices to convince ourselves that player i could release all of her packets at time $\sum_{k=1}^{i-1} q_k$ and follow the flow pattern of OPT_i without ever being delayed by a packet of players higher in the priority list.

For the sake of contradiction, we assume that a packet ℓ_i of player i has to wait for a packet ℓ_j of player $j < i$. If this is the case, there must exist an edge $e = (v, w)$ that is traversed by both packets ℓ_i and ℓ_j (see Figure 12). Hence, packet ℓ_j could have started at time $\sum_{k=1}^{i-1} q_k$ and arrive at node v at the same time as before, by taking the same s-v-path as packet ℓ_i. By the induction hypothesis, the original arrival time of packet ℓ_j is smaller or equal to $S_j - 1 + \sum_{k=1}^{i-1} q_k$. Note that if packet ℓ_j takes the same s-v-path as packet ℓ_i, and after that continues with its original v-t_j route, it leaves s after time $\sum_{k=1}^{i-1} q_k$ and arrives at t_j before time $\left(S_j - 1 + \sum_{k=1}^{i-1} q_k\right)$. Thus, the time that packet ℓ_j is in the network is bounded from above by:

$$\left(S_j - 1 + \sum_{k=1}^{i-1} q_k\right) - \sum_{k=1}^{i-1} q_k = S_j - 1.$$

This contradicts the fact that S_j is the length of a shortest s-t_j-path. Thus, all packets of player i can leave s at time $\sum_{j=1}^{i-1} q_j$, and arrive at t_i using their optimal strategy, without being delayed by previous players. Therefore:

$$C_{i,\ell}(NE) \leq C_{i,\ell}(OPT_i) + \sum_{k=1}^{i-1} q_k.$$

Further, no packet of player i arrives later than $S_i - 1 + \sum_{j=1}^{i-1} q_j$ since any best response of a player is an earliest arrival flow by Theorem 4. Thus, in no best response a packet falls behind a realizable time. ◀

▶ **Corollary 15.** *For a single source competitive packet routing game $\mathcal{G}$ with n players, demands $(k_i)_{i\in N}$ and arrival spreads $(q_i)_{i\in N}$ of the associated earliest arrival flows OPT_i for each $i \in N$, we have*

$$PoA(\mathcal{G}) \leq 1 + \frac{\sum_{i\in N} \sum_{j=i+1}^{n} q_i k_j}{\sum_{i\in N} C_i(OPT_i)}.$$

Proof. Assume that strategy OPT is a strategy that minimizes the social cost function. Using Theorem 14 we obtain that for any Nash equilibrium NE, we have that:

$$\frac{C(NE)}{C(OPT)} \leq \frac{\sum_{i\in N} C_i(OPT_i) + \sum_{i\in N} \sum_{j=i+1}^{n} q_i k_j}{\sum_{i\in N} C_i(OPT_i)} = 1 + \frac{\sum_{i\in N} \sum_{j=i+1}^{n} q_i k_j}{\sum_{i\in N} C_i(OPT_i)}. \quad ◀$$

We prove that this bound is actually tight.

▶ **Theorem 16.** *Let N be a set of n players and let $(q_i)_{i\in N}$ and $(k_i)_{i\in N}$ be arbitrary, but fixed, sequences of non-negative integers such that $q_i \le k_i$ for all $i \in N$. Then, there exists a single source competitive packet routing game $\tilde{\mathcal{G}}$ on n players with*

$$PoA(\tilde{G}) = 1 + \frac{\sum_{i\in N}\sum_{j=i+1}^{n} q_i k_j}{\sum_{i\in N} C_i(OPT_i)} = 1 + \frac{\sum_{i\in N}\sum_{j=i+1}^{n} q_i k_j}{\sum_{i\in N}\sum_{j=1}^{q_i} a_{i,j}(S_i + j - 1)}.$$

The proof of this theorem can be found in Appendix E. In the rest of this paper we create an algorithm that, for any set of players N with demands $(k_i)_{i\in N}$, can find arrival patterns for each player that maximizes the price of anarchy. First note that our goal is to maximize q_i while minimizing $C_i(OPT_i) = \sum_{j=1}^{q_i} a_{i,j}(S_i + j - 1)$.

▶ **Lemma 17.** *For any player with k_i packets and arrival spread of $q_i \le k_i$, her cost $C_i(OPT_i) = \sum_{j=1}^{q_i} a_{i,j}(S_i + j - 1)$ is minimized by $Q_i(S_i, q_i)$, where $Q_i(S_i, 1) := k_i S_i$ and*

$$Q_i(S_i, q_i) := k_i(S_i - 1) + \left\lfloor \frac{k_i - 1}{q_i - 1} \right\rfloor \cdot \frac{1}{2} q_i(q_i - 1) + \sum_{j=q_i-(k_i-1) \mod (q_i-1)}^{q_i} j.$$

Proof. Given a number of packets k_i and an arrival spread q_i, we determine the arrival pattern $(a_{i,p})_{p\le q_i}$ such that $\sum_{j=1}^{q_i} a_{i,j}(S_i + j - 1)$ is minimized. In order to get a feasible arrival pattern we are restricted to arrival patterns where $a_{i,1} \le \dots \le a_{i,q_i-1}$. We choose $a_{q_i} = 1$, and divide the $k_i - 1$ leftover packets evenly over the $q_i - 1$ leftover arrival times such that $a_{i,1} \le \dots \le a_{i,q_i-1}$. Thus:

$$a_{i,1} = \dots = a_{i,p} = \left\lfloor \frac{k_i - 1}{q_i - 1} \right\rfloor, \quad a_{i,p+1} = \dots = a_{i,q_i-1} = \left\lfloor \frac{k_i - 1}{q_i - 1} \right\rfloor + 1, \quad a_{i,q_i} = 1,$$

where $p = q_i - 1 - ((k_i - 1) \mod (q_i - 1))$. The total cost that corresponds to this arrival pattern is the $Q_i(S_i, q_i)$ described in the lemma. ◀

In order to find an example the expression mentioned in Theorem 16, we pick $S_i = 1$ and define $Q_i(q_i) := Q_i(1, q_i)$ for each $i \in N$. Then, we use Lemma 17 and it is left to maximize

$$P((q_i)_{i\in N}) := \frac{\sum_{i\in N}\left(Q_i(q_i) + q_i \sum_{j=i+1}^{n} k_j\right)}{\sum_{i\in N} Q_i(q_i)}. \tag{3}$$

Thus, in order to find an example that maximizes the price of anarchy, we only need to decide on a q_i for each player. In order to do so, we define $\mu_{i,OPT}(p) := Q_i(p+1) - Q_i(p)$ and $\mu_{i,NE}(p) := Q_i(p+1) - Q_i(p) + \sum_{j=i+1}^{n} k_j$. Intuitively, if a player decides to increase q_i from p to $p+1$, it would add a cost of $\mu_{i,OPT}(p)$ to the social optimum and a cost of $\mu_{i,NE}(p)$ to the worst equilibrium. We state Algorithm 1 using $\mu_{i,OPT}(p)$ and $\mu_{i,NE}(p)$.

▶ **Theorem 18.** *Given a set of players N, where each player has a demand k_i. Then, Algorithm 1 returns an sequence $q := (q_i)_{i\in N}$ that maximizes $P(q)$ as defined in* (3).

Proof. First note, that by definition

$$\frac{\sum_{i\in N}\sum_{p=0}^{q_i-1} \mu_{i,NE}(p)}{\sum_{i\in N}\sum_{p=0}^{q_i-1} \mu_{i,OPT}(p)} = \frac{\sum_{i\in N}\left(Q_i(q_i) + q_i \sum_{j=i+1}^{n} k_j\right)}{\sum_{i\in N} Q_i(q_i)}.$$

Let $q' \in \arg\max P(q)$ and let $(q_i)_{i\in N}$ be the output of Algorithm 1. Assume $(q_i)_{i\in N}$ is not optimal, thus $P(q') > P(q)$. We first prove that $q'_i \le q_i$ for all $i \in N$.

We define q_{-i} to be the vector $(q_j)_{j\in N\setminus\{i\}}$. For sake of contradiction, assume that there exists an $i \in N$ such that $q'_i > q_i$. We distinguish two cases:

Algorithm 1: Creating an example with maximized price of anarchy.

Input: A set N consisting of n players with k_i packets.
Output: A vector $(q_i)_{i \in N}$.

1 $q_i \leftarrow 1$ for all $i \in N$;
2 **for** $i \in N$ **do**
3 $\quad p_i \leftarrow \arg\max_{q_i \leq p < k_i} \left\{ \frac{\sum_{q=q_i}^{p} \mu_{i,NE}(q)}{\sum_{q=q_i}^{p} \mu_{i,OPT}(q)} \right\}$;
4 $\quad P_i \leftarrow \max_{q_i \leq p < k_i} \left\{ \frac{\sum_{q=q_i}^{p} \mu_{i,NE}(q)}{\sum_{q=q_i}^{p} \mu_{i,OPT}(q)} \right\}$;
5 **end**
6 $j \leftarrow \arg\max_{i \in N}\{P_i\}$;
7 **while** $P_j > \frac{\sum_{i \in N}\left(Q_i(q_i) + q_i \sum_{j=i+1}^{n} k_j\right)}{\sum_{i \in N} Q_i(q_i)}$ **do**
8 $\quad q_j \leftarrow p_j + 1$;
9 $\quad p_j \leftarrow \arg\max_{q_j \leq p < k_j} \left\{ \frac{\sum_{q=q_j}^{p} \mu_{j,NE}(q)}{\sum_{q=q_j}^{p} \mu_{j,OPT}(q)} \right\}$;
10 $\quad P_j \leftarrow \max_{q_j \leq p < k_j} \left\{ \frac{\sum_{q=q_j}^{p} \mu_{j,NE}(q)}{\sum_{q=q_j}^{p} \mu_{j,OPT}(q)} \right\}$;
11 $\quad j \leftarrow \arg\max_{i \in N}\{P_i\}$;
12 **end**
13 **return** $(q_i)_{i \in N}$

1. $P(q_{-i}, q_i') > P(q)$. In this case the algorithm would not terminate. From the assumption $P(q_{-i}, q_i') > P(q)$ we get that
$$\frac{\sum_{i \in N} \sum_{p=0}^{q_i - 1} \mu_{i,NE}(p) + \sum_{p=q_i}^{q_i'-1} \mu_{i,NE}(p)}{\sum_{i \in N} \sum_{p=0}^{q_i - 1} \mu_{i,OPT}(p) + \sum_{p=q_i}^{q_i'-1} \mu_{i,OPT}(p)} > \frac{\sum_{i \in N} \sum_{p=0}^{q_i - 1} \mu_{i,NE}(p)}{\sum_{i \in N} \sum_{p=0}^{q_i - 1} \mu_{i,OPT}(p)}.$$
Thus,
$$\frac{\sum_{p=q_i}^{q_i'-1} \mu_{i,NE}(p)}{\sum_{p=q_i}^{q_i'-1} \mu_{i,OPT}(p)} > P(q),$$
is one candidate for P_j determined in line 10 of the algorithm. This candidate is already larger than $P(q)$, thus the algorithm would not terminate with q respectively $P(q)$. Thus, this contradicts the fact that Algorithm 1 outputs q.
2. $P(q_{-i}, q_i') \leq P(q)$. In this case, $(\sum_{p=q_i}^{q_i'-1} \mu_{i,NE}(p))/(\sum_{p=q_i}^{q_i'-1} \mu_{i,OPT}(p)) \leq P(q) < P(q')$. Decreasing q_i' to q_i would increase the quotient of $P(q')$, which means $P(q_{-i}', q_i) > P(q')$. This is a contradiction to $q' \in \arg\max P(q)$.

Thus, we have shown that if q is not optimal, $q_i' \leq q_i$ for all $i \in N$. It remains to show that $q_i' < q_i$ leads to a contradiction. Due to the initialization of $q_i = 1$ which is minimal, we know that q_i is less or equal to q_i' at the start of Algorithm 1. Assume that during the execution of Algorithm 1, we obtain the following vectors for $(q_i)_{i \in N}$: $\vec{1}, q^1, \ldots, q^k, q$.

If there is a $i \in N$ such that $q_i' < q_i$, during Algorithm 1 there needs to be a vector q^b such that $q_j^b \leq q_j'$ for all $j \in N$ and there is an $i \in N$ such that $q_i^{b+1} > q_i'$ and $q_j^{b+1} \leq q_j'$ for all $j \in N \backslash \{i\}$. This means q^{b+1} is the vector where for the first time in Algorithm 1 a value of q is increased over a value of q'. By definition of Algorithm 1, we know that the chosen value q_i^{b+1} maximizes the quotient of marginal cost increase of Nash equilibrium over optimal solution among all alternative vectors. This means:

$$\frac{\sum_{p=q_i^b}^{q_i^{b+1}-1} \mu_{i,NE}(p)}{\sum_{p=q_i^b}^{q_i^{b+1}-1} \mu_{i,OPT}(p)} \geq \max_{j\in N} \frac{\sum_{p=q_j^b}^{q_j'-1} \mu_{j,NE}(p)}{\sum_{p=q_j^b}^{q_j'-1} \mu_{j,OPT}(p)}. \tag{4}$$

Furthermore, by the choice of q' we know that:

$$P(q') = \frac{\sum_{i\in N}\left(\sum_{p=0}^{q_i^b-1} \mu_{i,NE}(p) + \sum_{p=q_i^b}^{q_i'-1} \mu_{i,NE}(p)\right)}{\sum_{i\in N}\left(\sum_{p=0}^{q_i^b-1} \mu_{i,OPT}(p) + \sum_{p=q_i^b}^{q_i'-1} \mu_{i,OPT}(p)\right)}.$$

By definition of Algorithm 1:

$$\frac{\sum_{p=q_i^b}^{q_i^{b+1}-1} \mu_{i,NE}(p)}{\sum_{p=q_i^b}^{q_i^{b+1}-1} \mu_{i,OPT}(p)} > P(q^b) = \frac{\sum_{i\in N}\sum_{p=0}^{q_i^b-1} \mu_{i,NE}(p)}{\sum_{i\in N}\sum_{p=0}^{q_i^b-1} \mu_{i,OPT}(p)}. \tag{5}$$

and by (4):

$$\frac{\sum_{p=q_i^b}^{q_i^{b+1}-1} \mu_{i,NE}(p)}{\sum_{p=q_i^b}^{q_i^{b+1}-1} \mu_{i,OPT}(p)} \geq \frac{\sum_{i\in N}\sum_{p=q_i^b}^{q_i'-1} \mu_{i,NE}(p)}{\sum_{i\in N}\sum_{p=q_i^b}^{q_i'-1} \mu_{i,OPT}(p)}. \tag{6}$$

Given $\frac{a_1}{a_2}, \frac{b_1}{b_2}, \frac{c_1}{c_2} \in \mathbb{Q}$, then, whenever $\frac{a_1}{a_2} > \frac{b_1}{b_2}$ and $\frac{a_1}{a_2} \geq \frac{c_1}{c_2}$, it holds that $\frac{a_1}{a_2} > \frac{b_1+c_1}{b_2+c_2}$. We use this type of argumentation on (5) and (6) to obtain:

$$\frac{\sum_{p=q_i^b}^{q_i^{b+1}-1} \mu_{i,NE}(p)}{\sum_{p=q_i^b}^{q_i^{b+1}-1} \mu_{i,OPT}(p)} > P(q').$$

Hence, one could increase $P(q')$ by increasing q_i' to q_i^{b+1}. This contradicts the fact that q' maximizes $P(\cdot)$. ◀

▶ Remark. The running time of the algorithm is polynomial in $k_1, \ldots, k_n$ and n.

Proof. In the initial phase we compute n times a maximum value which takes at most k_i^3 time for every $i \in N$. In the while loop, we do the same computation. Note that in every execution of the while loop one q_i for $i \in N$ is increased by at least one. Since the q_i's are initialized as one and bounded from above by k_i, the while loop takes at most $\sum_{i\in N} k_i$ iterations. Hence, the running time of the algorithm is polynomial in $(k_i)_{i\in N}$ and n. ◀

In Appendix F we apply this algorithm to a small example.

References

1 Elliot Anshelevich, Anirban Dasgupta, Jon Kleinberg, Eva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing*, 38(4):1602–1623, 2008.

2 Nadine Baumann and Martin Skutella. Earliest arrival flows with multiple sources. *Mathematics of Operations Research*, 34(2):499–512, 2009.

3 Lisa K Fleischer. Faster algorithms for the quickest transshipment problem. *SIAM journal on Optimization*, 12(1):18–35, 2001.

4 Lisa K Fleischer and Martin Skutella. Quickest flows over time. *SIAM Journal on Computing*, 36(6):1600–1630, 2007.

5 David Gale. Transient flows in networks. *Michigan Math. J.*, 6(1):59–63, 1959.

6 Tobias Harks, Britta Peis, Daniel Schmand, Bjoern Tauer, and Laura Vargas Koch. Competitive packet routing with priority lists. *ACM Trans. Econ. Comput.*, 6(1):4:1–4:26, March 2018.

7 David G. Harris and Aravind Srinivasan. Constraint satisfaction, packet routing, and the lovasz local lemma. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 685–694, New York, NY, USA, 2013. ACM.

8 Martin Hoefer, Vahab S Mirrokni, Heiko Röglin, and Shang-Hua Teng. Competitive routing over time. *Theoretical Computer Science*, 412(39):5420–5432, 2011.

9 John J Jarvis and H Donald Ratliff. Note - some equivalent objectives for dynamic network flow problems. *Management Science*, 28(1):106–109, 1982.

10 Ronald Koch, Britta Peis, Martin Skutella, and Andreas Wiese. Real-time message routing and scheduling. In *Approx., Rand., and Comb. Opt. Algorithms and Techniques*, pages 217–230. Springer Berlin Heidelberg, 2009.

11 Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. In *Stacs*, volume 99, pages 404–413. Springer, 1999.

12 Janardhan Kulkarni and Vahab Mirrokni. Robust price of anarchy bounds via lp and fenchel duality. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1030–1049. SIAM, 2014.

13 Frank Thomson Leighton, Bruce M Maggs, and Satish B Rao. Packet routing and job-shop scheduling in $\mathcal{O}$(congestion + dilation) steps. *Combinatorica*, 14(2):167–186, 1994.

14 Nimrod Megiddo. Optimal flows in networks with multiple sources and sinks. *Mathematical Programming*, 7(1):97–107, 1974.

15 Edward Minieka. Maximal, lexicographic, and dynamic network flows. *Operations Research*, 21(2):517–527, 1973.

16 Britta Peis and Andreas Wiese. Universal packet routing with arbitrary bandwidths and transit times. In *Proceedings of the 15th International Conference on Integer Programming and Combinatoral Optimization*, IPCO'11, pages 362–375, Berlin, Heidelberg, 2011. Springer-Verlag.

17 Thomas Rothvoß. A simpler proof for o(congestion + dilation) packet routing. *CoRR*, abs/1206.3718, 2012.

18 Tim Roughgarden. Routing games. In *Algorithmic Game Theory*, pages 461–486. Cambridge Univ. Press, 2007.

19 Christian Scheideler and Berthold Vöcking. From static to dynamic routing: Efficient transformations of store-and-forward protocols. *SIAM Journal on Computing*, 30(4):1126–1155, 2000.

20 Martin Skutella. An introduction to network flows over time. In *Research trends in combinatorial optimization*, pages 451–482. Springer, 2009.

21 Aravind Srinivasan and Chung-Piaw Teo. A constant-factor approximation algorithm for packet routing and balancing local vs. global criteria. *SIAM Journal on Computing*, 30(6):2051–2068, 2001.

22 Stevanus Adrianto Tjandra. *Dynamic network optimization with application to the evacuation problem.* Shaker, 2003.

23 William L Wilkinson. An algorithm for universal maximal dynamic flows in a network. *Operations Research*, 19(7):1602–1612, 1971.

A Technical details of the Proof of Theorem 4

We use ideas of [6] to prove that:

$$\min \sum_{\ell \in K_i} C_{i,\ell}(x) = \max \sum_{\theta \in \mathbb{N}_{>0}} A_i(x, \theta - 1).$$

Observe that:

$$\begin{aligned}
& \sum_{\ell \in K_i} C_{i,\ell}(x) \\
= & \sum_{\theta \in \mathbb{N}_{>0}} \left(A_i(x,\theta) - A_i(x,\theta-1)\right)\theta \\
= & \sum_{\theta \in \mathbb{N}_{>0}} A_i(x,\theta)\theta - \sum_{\theta \in \mathbb{N}_{>0}} A_i(x,\theta-1)\theta \\
= & \sum_{\theta \in \mathbb{N}_{>0}} A_i(x,\theta)\theta - \sum_{\theta \in \mathbb{N}_{>0}} A_i(x,\theta-1)(\theta-1) - \sum_{\theta \in \mathbb{N}_{>0}} A_i(x,\theta-1).
\end{aligned}$$

Note that $\sum_{\theta \in \mathbb{N}_{>0}} A_i(x,\theta)\theta = \sum_{\theta \in \mathbb{N}_{>0}} A_i(x,\theta-1)(\theta-1)$, as $A_i(x,0) = 0$. Thus:

$$\min \sum_{\ell \in K_i} C_{i,\ell}(x) = \min - \left(\sum_{\theta \in \mathbb{N}_{>0}} A_i(x,\theta-1) \right) = \max \sum_{\theta \in \mathbb{N}_{>0}} A_i(x,\theta-1).$$

B Cycle free path decomposition

▶ **Lemma 19.** *For any s-t-graph G, there exist an earliest arrival flow for varying capacities that has a path decomposition where no flow is send along cycles.*

Proof of Lemma 19. First we construct an earliest arrival flow by using the algorithm of Tjandra [22]. The algorithm is roughly speaking a successive shortest path algorithm in a network with varying capacities. We prove that there exists a sequence of shortest paths in the successive shortest path algorithm such that the resulting flow does not contain cycles. If no cycles occurs in the flow, then no cycles occur in any path decomposition of the earliest arrival flow.

During the successive shortest path algorithm, cycles can arise in two different ways.

1. During the course of the algorithm, we choose a shortest path that contains a cycle. As all transit times are non-negative, the length of the cycle is bounded from below by zero. Hence, we can delete this cycle and use the resulting (shortest) path.
2. During the course of the algorithm, we add a shortest path P that closes a directed cycle for the first time, by connecting some nodes u and v by forward edges. Thus, there needs to be a directed sequence of edges connecting the nodes v and u. Instead of closing the cycle, the path could also go along this forward edges as backwards edges. Since the cost of the sequence of forward edges is lower bounded by zero and the cost of the backwards edges is upper bounded by zero, this never increases the costs of the path. Thus, this is a feasible choice for a shortest path.

Hence, there exists a sequence of shortest paths such that the resulting flow does not contain any cycles. ◀

C Proof of Lemma 8

Proof of Lemma 8. We prove this lemma by induction. Note that, as player 1 is not affected by other players, $C_{1,\ell}(NE) = C_{1,\ell}(OPT)$. Hence, the lemma clearly holds for the first player.

Assume that the lemma holds for the first $i-1$ players (players with highest priority) then we prove that $C_{i,\ell}(NE) \leq i \cdot C_{i,\ell}(OPT)$. As player i comes after player j on the priority list for any player $j < i$, we have, by construction of OPT, that $C_{i,\ell}(OPT) \geq C_{j,\ell}(OPT)$.

Hence:

$$\frac{C_{j,\ell}(NE)}{C_{i,\ell}(OPT)} \leq \frac{C_{j,\ell}(NE)}{C_{j,\ell}(OPT)} \leq j,$$

where the last inequality holds as of our induction hypothesis. Therefore, we know that:

$$C_{j,\ell}(NE) \leq j \cdot C_{i,\ell}(OPT). \tag{7}$$

Observe that in the worst case, player i can play the same strategy as she did in the optimal solution, but only after all previous players $j < i$ have already left the network. Hence:

$$C_{i,\ell}(NE) \leq \max_{j<i,\ell \in K_j} \{C_{j,\ell}(NE)\} + C_{i,\ell}(OPT). \tag{8}$$

We combine inequalities (7) and (8) to obtain

$$C_{i,\ell}(NE) \leq \max_{j<i,\ell \in K_j} \{j \cdot C_{i,\ell}(OPT)\} + C_{i,\ell}(OPT).$$

Then, $j \cdot C_{i,\ell}(OPT)$ is clearly maximized whenever $j = i - 1$. Hence, we obtain:

$$C_{i,\ell}(NE) \leq (i-1) \cdot C_{i,\ell}(OPT) + C_{i,\ell}(OPT) = i \cdot C_{i,\ell}(OPT),$$

which proves the lemma. ◀

D Details of Theorem 11

In this appendix, one can find the technical details of the proof of Theorem 11. We formally prove why:

$$PoA \leq \frac{\sum_{i=1}^{n} \sum_{p=1}^{q_1} a_p(S + p - 1 + (i-1)q_1)}{knS + \sum_{p=1}^{q_1} a_p(p-1) + (q_1 - 1)k(n-1)} \leq \frac{1}{2}(n+1).$$

Technical details. During the proof of Theorem 11 we established a lower bound on the cost of a socially optimal profile:

$$C(OPT) \geq knS + \sum_{p=1}^{q_1} a_p(p-1) + (q_1 - 1)k(n-1).$$

Furthermore, we bounded the cost of any equilibrium from above by:

$$\begin{aligned} C(NE) &\leq \sum_{i=1}^{n} \sum_{p=1}^{q_1} a_p(S + p - 1 + (i-1)q_1) \\ &= n\left(\sum_{p=1}^{q_1} a_p(S + p - 1)\right) + \tfrac{1}{2}q_1 kn(n-1) \\ &= knS + n\left(\sum_{p=1}^{q_1} a_p(p-1)\right) + \tfrac{1}{2}q_1 kn(n-1). \end{aligned}$$

As we have a lower bound on the social cost, and an upper bound of the cost in any Nash equilibrium, we can find an upper bound on the price of anarchy.

$$PoA \leq \frac{knS + n\left(\sum_{p=1}^{q_1} a_p(p-1)\right) + \frac{1}{2}q_1kn(n-1)}{knS + \sum_{p=1}^{q_1} a_p(p-1) + (q_1-1)k(n-1)}$$

Note that for all $n \geq 1, k \geq 1$, we have that

$$n\left(\sum_{p=1}^{q_1} a_p(p-1)\right) + \frac{1}{2}q_1kn(n-1) \geq \left(\sum_{p=1}^{q_1} a_p(p-1)\right) + (q_1-1)k(n-1) \geq 0.$$

As $k, n, S \geq 1$, the PoA is maximized when knS is minimal, which is the case when $S = 1$. We obtain:

$$\begin{aligned} PoA &\leq \frac{kn + n\left(\sum_{p=1}^{q_1} a_p(p-1)\right) + \frac{1}{2}q_1kn(n-1)}{kn + \sum_{p=1}^{q_1} a_p(p-1) + (q_1-1)k(n-1)} \\ &= \frac{n\left(\sum_{p=1}^{q_1} pa_p\right) + \frac{1}{2}q_1kn(n-1)}{\sum_{p=1}^{q_1} pa_p + q_1k(n-1)} \\ &= \frac{n\left(\sum_{p=1}^{q_1} pa_p + q_1k(n-1)\right) - \frac{1}{2}kq_1n(n-1)}{\sum_{p=1}^{q_1} pa_p + q_1k(n-1)} \\ &= n - \frac{\frac{1}{2}q_1kn(n-1)}{\sum_{p=1}^{q_1} pa_p + q_1k(n-1)}. \end{aligned}$$

Thus, it is left to minimize $\left(\frac{1}{2}q_1kn(n-1)\right) / \left(\sum_{p=1}^{q_1} pa_p + q_1k(n-1)\right)$. Note that for any q_1, $\sum_{p=1}^{q_1} pa_p$ is maximized when $a_p = 1$ for $p \in \{1, \dots, q_1-1\}$ and $a_{q_1} = k - q_1 + 1$. Hence, for any q_1,

$$\sum_{p=1}^{q_1} pa_p \leq \frac{1}{2}q_1(q_1-1) + q_1(k-q_1+1) = q_1(k - \frac{1}{2}(q_1-1)).$$

Thus,

$$\begin{aligned} n - \frac{\frac{1}{2}q_1kn(n-1)}{\sum_{p=1}^{q_1} pa_p + q_1k(n-1)} &\leq n - \frac{\frac{1}{2}q_1kn(n-1)}{q_1(k-\frac{1}{2}(q_1-1)) + q_1k(n-1)} \\ &= n - \frac{\frac{1}{2}kn(n-1)}{kn - \frac{1}{2}(q_1-1)}. \end{aligned}$$

As $q_1 \geq 1$, the PoA is clearly maximized when $q_1 = 1$. We obtain:

$$PoA \leq n - \frac{\frac{1}{2}kn(n-1)}{kn - \frac{1}{2}(1-1)} = n - \frac{\frac{1}{2}kn(n-1)}{kn} = \frac{1}{2}(n+1),$$

which proves the theorem. ◀

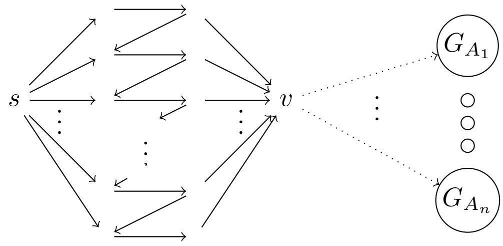

Figure 13 Braess graph BG with player specific paths to individual sink.

E Proof of Theorem 16

Proof. We denote the arrival pattern of a player i in an optimal solution where player i is the only player in the network by $A_i := (a_{i,1}, \dots, a_{i,q_i})$. Here $a_{i,p}$ denotes the number of packets that arrive at time $S_i + p - 1$. At time $S_i + q_i - 1$ the last packet of player i arrives, i.e. player i has an arrival spread of $q_i := \arg\min_{p \in \mathbb{N}_{>0}} \{a_{i,q'} = 0, \forall q' > 0\}$, thus, $\sum_{p=1}^{q_i} a_{i,p} = k_i$. Again note that $1 \leq a_{i,1} \leq a_{i,2} \leq \dots \leq a_{i,q_i-1}$. This holds true with a simple following argumentation. If p packets arrive at time θ, p further packets can arrive at time $\theta + 1$ by following the first p packets.

Given a set of players N with $(k_i)_{i\in N}$ and $(q_i)_{i\in N}$ with $k_i \geq q_i$, we can construct an arrival pattern $A_i := (a_{i,1}, \dots, a_{i,q_i})$ for every player i realizing her k_i and q_i in the following way:

$$a_{i,1} = \dots = a_{i,p} = \left\lfloor \frac{k_i - 1}{q_i - 1} \right\rfloor, \quad a_{i,p+1} = \dots = a_{i,q_i-1} = \left\lfloor \frac{k_i - 1}{q_i - 1} \right\rfloor + 1, \quad a_{i,q_i} = 1,$$

where $p = q_i - 1 - ((k_i - 1) \mod (q_i - 1))$.

For every player i with arrival pattern $A_i := (a_{i,p})_{1 \leq p \leq q_i}$, we construct a corresponding s_i-t_i-graph $G_{A_i} = (V_{A_i}, E_{A_i})$ consisting of only parallel s_i-t_i-edges, such that the arrival pattern of the earliest arrival flow of G_{A_i} matches A_i. In order to do so, we first define $a_{i,0} = 0$. Then, we add $\max\{a_{i,p} - a_{i,p-1}, 0\}$ parallel edges of length $S + p - 1$ and capacity one for all $1 \leq p \leq q_i$ to graph G_{A_i} for all $i \in N$.

We define $K := \sum_{i\in N} k_i$, and define $BG(K)$ as in Section 2. We connect B_K and G_{A_i} by setting $v \in V_{BG(K)}$ equal to $s_i \in V_{A_i}$ for all $i \in N$ as in Figure 13.

In a socially optimal solution, each player $i \in N$ can enter their graph G_{A_i} at time zero, and thus arrive at sink t_i according to arrival pattern A_i, resulting in a social cost $\sum_{i\in N} C_i(OPT_i)$. In the worst Nash equilibrium, player i blocks graph $BG(K)$ for q_i units of time, delaying all players $j > i$ by q_i time units. This results in a total cost of $\sum_{i\in N} \left(C_i(OPT_i) + \sum_{j=i+1}^{n} q_i k_j \right)$. As this holds for all players $i \in N$, this gives us the desired price of anarchy. ◀

F Example algorithm 1

▶ **Example 20.** For a better understanding of the algorithm we apply it to a small example. We are given two players with a demand of four each, and we return a game that maximizes the price of anarchy for the given demands. We start with $q_1 = q_2 = 1$. In the for loop of Algorithm 1 p_i and P_i are determined. We start with player 1: for increasing q_1 from 1 to 2, we get a quotient of $\frac{7}{3}$, for increasing it to 3 we get $\frac{12}{4} = 3$ and for increasing it from 1

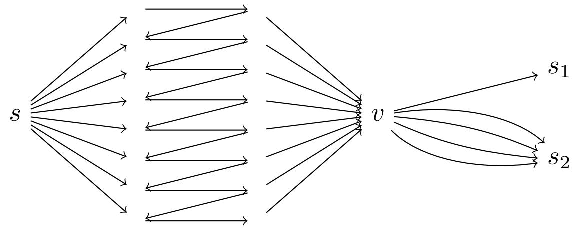

Figure 14 Example with price of anarchy of $\frac{30}{14}$.

to 4 we get $\frac{18}{6} = 3$. Thus, we obtain $(p_1, P_1) = (3, \frac{12}{4})$. Similarly, for player 2 we obtain $(p_2, P_2) = (2, 1)$. Therefore, after the first for loop we choose $j \leftarrow 1$.

Since $\frac{12}{4} > \frac{4+4+4}{4+4}$ we enter into the `while` loop. We increase q_1 from 1 to 3. and update the values of p_1 and P_1, $(p_1, P_1) \leftarrow (4, \frac{6}{2})$. Hence, again $j \leftarrow 1$.

Since $\frac{6}{2} > \frac{24}{12}$ we enter the `while` loop a second time. We set $q_1 = 4$ and update p_1 and P_1. Since q_1 cannot be increased, $P_1 = 0$. Hence, $j \leftarrow 2$.

Since $\frac{1}{1} > \frac{30}{14}$ is not correct, we do not enter the `while` loop again and return $q = (4, 1)$. This results in a price of anarchy of $\frac{30}{14}$. Note that this is larger than two and thus strictly worse than in the single commodity case, where we established an upper bound of n. The graph realizing this price of anarchy is depicted in Figure 14.

In the optimal solution, the arrival times of player 1 are $1, 2, 3, 4$, and for player 2 we obtain $1, 1, 1, 1$, resulting in a total cost of 14. In the the worst Nash equilibrium, the arrival times of player 1 are $1, 2, 3, 4$, and the arrival times of player 2 are $5, 5, 5, 5$, resulting in a total cost of 30. Hence, the price of anarchy is indeed $\frac{30}{14}$.