Routing Games with Progressive Filling

Tobias Harks*, Martin Hoefer†, Kevin Schewior‡, Alexander Skopalik§
*Dept. of Quantitative Economics, Maastricht University, Email: t.harks@maastrichtuniversity.nl
†Max-Planck-Institut für Informatik and Saarland University, Email: mhoefer@mpi-inf.mpg.de
‡Institut für Mathematematik, TU Berlin, Email: schewior@math.tu-berlin.de
§Dept. of Computer Science, Paderborn University, Email: skopalik@mail.upb.de

***Abstract*—Max-min fairness (MMF) is a widely known approach to a fair allocation of bandwidth to each of the users in a network. This allocation can be computed by uniformly raising the bandwidths of all users without violating capacity constraints. We consider an extension of these allocations by raising the bandwidth with arbitrary and not necessarily uniform time-depending velocities (allocation rates). These allocations are used in a game-theoretic context for routing choices, which we formalize in progressive filling games (PFGs).**

We present a variety of results for equilibria in PFGs. We show that these games possess pure Nash and strong equilibria. While computation in general is NP-hard, there are polynomial-time algorithms for prominent classes of Max-Min-Fair Games (MMFG), including the case when all users have the same source-destination pair. We characterize prices of anarchy and stability for pure Nash and strong equilibria in PFGs and MMFGs when players have different or the same source-destination pairs. In addition, we show that when a designer can adjust allocation rates, it is possible to design games with optimal strong equilibria. Some initial results on polynomial-time algorithms in this direction are also derived.

Keywords*—*routing, congestion control, existence of strong and Nash equilibrium, complexity and convergence, price of anarchy

I. Introduction

Max-min fairness is a widely used paradigm for bandwidth allocation problems in telecommunication networks, most prominently, it is used as a reference point for designing flow control/congestion control protocols such as TCP (Transport Control Protocol), see [25] for a more detailed discussion. In a max-min fair allocation, the bandwidth of a user cannot be increased without decreasing the bandwidth of another user, who already receives a smaller bandwidth. Max-min fairness also plays an important role in the model of Kelly et al. [14], where congestion control protocols have been interpreted as distributed algorithms at sources and links in order to solve a global optimization problem (cf. [17], [16], [21] for further works in this area). Each user is associated with an increasing, strictly concave bandwidth utility function and the congestion control algorithms aim at maximizing aggregate utility subject to capacity constraints on the links. Mo and Walrand [21] showed that within the model of Kelly et al., there is a family of utility functions whose global optimum corresponds to a max-min fair bandwidth allocation and they devised a distributed max-min fair congestion control protocol, see also [24] (Section 2.2) and [19]. For further distributed max-min fair congestion control protocols, we refer to [28], [30]. There are several important generalizations of max-min fairness such as weighted max-min fairness [28] and utility max-min fairness [8]. In a weighted max-min fair allocation, the weighted bandwidth of a user cannot be increased without decreasing the weighted bandwidth of another user, who already receives a smaller weighted bandwidth. In a utility max-min fair allocation, each user is associated with an increasing (not necessarily concave) bandwidth utility function and an allocation is utility max-min fair if the utility of a user cannot be increased without decreasing the utility of another user, who already receives a smaller utility. Utility max-min fairness (and also weighted max-min fairness) has been proposed for giving some applications (e.g., real-time applications, or multi-media) a possibly larger bandwidth share than others.

It is well known that (weighted) max-min fair allocations can be easily implemented by simple polynomial time water-filling algorithms that raise the bandwidth of every user at a (weighted) uniform speed and, whenever a link capacity is exhausted, fixes the bandwidth of those users traversing this link [4]. As we will show in this paper, also utility max-min fair allocations can be implemented by simple polynomial time water-filling algorithms that raise the bandwidth of every user at a user-specific speed.

While most works in the area of flow control/congestion control assume that the routes of users are fixed a priori, we study in this paper the flexibility of *strategic route choices* by users (or players from now on) as a means to obtain high bandwidth. We introduce a general class of strategic games that we term *routing games with progressive filling*. In such a game, there is a finite set of resources and a strategy of a player corresponds to a subset of resources. Resources have capacities and the utility of every player equals the obtained bandwidth which in turn is defined by a predefined water-filling algorithm. If the allowable subsets of a player correspond to the set of routes connecting the player's source with its terminal, we obtain single-path routing modeling IP (Internet Protocol) routing. Since IP routing is typically updated at a much slower timescale than the flow control, we assume that flow control (modeled in this paper as a water-filling algorithm) converges instantly to a "fair" allocation (max-min fair or generalizations thereof) after each route update. The assumption that flow control converges instantly before route updates are triggered has been made and justified before, see, e.g., Wang et al. [26]. Thus, once a player chooses a new route his bandwidth share

This research was partially supported by the German Research Foundation (DFG) within the Cluster of Excellence MMCI at Saarland University, the Research Training Group "Methods for Discrete Structures" (GRK 1408) and the Collaborative Research Center "On-The-Fly Computing" (SFB 901). It was also partially supported by the EU within FET project MULTIPLEX (contract no. 317532) and the Marie-Curie grant "Protocol Design" (no. 327546, funded within FP7-PEOPLE-2012-IEF).

is determined by executing the water-filling algorithm. We will impose mild conditions on the class of allowable water-filling algorithms: (i) for every player and every point in time the integral of the rate function is non-negative and the integral of the rate function grows monotonically; (ii) for every player the integral of the rate function tends to infinity as time goes to infinity. While condition (i) is natural, condition (ii) simply ensures that the water filling algorithm terminates and the induced strategic game is well-defined. Note that even though water-filling algorithms are *centralized* algorithms we demonstrate that they represent a wide range of fairness concepts including max-min fairness, weighted max-min fairness and utility max-min fairness for which *distributed* and *fast converging* congestion control protocols are known [20], [21], [28], [30].

We consider existence, computation and quality of equilibria in routing games with progressive filling. In a pure Nash equilibrium (PNE for short), no player obtains strictly higher bandwidth by unilaterally changing his route. If coordinated deviations by players are allowed (for instance by a single player coordinating several sessions or by a set of players connected via peer-to-peer overlay networks), the Nash equilibrium concept is not sufficient to analyze stable states of a game. For this situation, we adopt the stricter notion of a strong equilibrium (SE for short) proposed by Aumann. In a SE, no coalition (of any size) can change their routes and strictly increase the bandwidth of each of its members (while possibly lowering the bandwidth of players outside the coalition). Every SE is a PNE, but not conversely. Thus, SE constitute a very robust and appealing stability concept for which only a few existence results are known in the literature.

A. *Our Results*

Existence. For progressive filling games we prove that if water-filling algorithms satisfy conditions (i) and (ii), every sequence of profitable deviations of coalitions of players must be finite and, hence, SE always exist. Previously, it was only known that PNE exist if the water-filling algorithm corresponds to the max-min fair allocation [29]. Thus, our results establish for the first time that routing and congestion control admits a PNE (and even SE) for routing games where weighted- and utility max-min fair congestion control protocols are used. We show that our assumptions (i) and (ii) are "minimal" in the sense that if one of them is dropped, there is a corresponding two-player game without PNE.

Complexity. In light of its practical importance, we study routing games with water-filling algorithms inducing the max-min fair allocation. We first focus on the computational complexity of SE and PNE. We give an algorithm that computes a SE for any progressive filling game under max-min fair allocations. Our algorithm iteratively reduces the number of players allowed on a resource. After each such reduction, a *packing oracle* is invoked that checks whether or not there is a feasible strategy profile that respects the allowed numbers of players on every resource. If the oracle finds a feasible allocation, the algorithm proceeds and, otherwise, we fix strategies for a suitable subset of players. Obviously, the running time of the algorithm crucially relies on the running time of the packing oracle. It is known, however, that if the strategy spaces correspond to, e.g., the set of paths of a single-commodity network, or to bases of a matroid defined on a player-specific subset of resources, the oracle can be implemented in polynomial time, thereby ensuring polynomial-time computation of SE. We complement this result by showing various hardness results of computing SE. In addition, we show a bound on the number of values of the potential function that also represents an upper bound on the number of improvement steps to reach a PNE.

Quality. To measure the quality of an equilibrium, we use the achieved throughput defined as the sum of the player's bandwidths. This performance measure corresponds to utilitarian social welfare and is the standard performance measure in traffic engineering. We use notions of price of stability (PoS) and price of anarchy (PoA), which relate the cost of an equilibrium to the cost of a social optimum. The standard definition of an optimum would refer to a set of route choices such that throughput is maximized for a waterfilling algorithm with given allocation rates. In addition, our bounds continue to hold even with respect to an optimum that is allowed to set arbitrary routes and bandwidths respecting the resource capacities. Computing this general optimum is known in combinatorial optimization as the maximum k-splittable flow problem.

We provide tight bounds for SE and PNE. In general, the PoS and PoA are n, which is tight for both PNE and SE, even in multi-commodity MMFGs. In single-commodity MMFGs, PoS for PNE and SE is $\left(2 - \frac{1}{n}\right)$, PoA for PNE is n and PoA for SE is 4. All bounds except the latter are tight. In addition, our algorithm that computes SE for single-commodity MMFGs in polynomial time yields SE that match the PoS bound.

Protocol Design. Using fixed allocation rates, improving upon the $\left(2 - \frac{1}{n}\right)$-bound is impossible in the worst case. We show, however, that it is possible to show better results when we have slight flexibility in allocation rates. We assume the freedom to "design a protocol" and adjust weights in a weighted MMF waterfilling algorithm towards the topology of the instance. This allows to design a game with an optimal SE that coincides with the maximum k-splittable flow. While computing such an optimum is NP-hard, the result also shows that starting from any α-approximation to the maximum k-splittable flow, we can design weights and a starting state, such that every sequence of unilateral (coalitional) improvement moves leads to a PNE (SE) with the same approximation ratio. We apply this approach in games with 3 players, where we can find in polynomial time a solution that is a 1.5-approximation and represents a PNE for the chosen weights. For spatial reasons, some of the proofs and results are deferred to the complete version of this paper available at arxiv.org.

B. *Related Work*

Combined routing and congestion control has been studied by several works (cf. [7], [27], [15], [13]). In all these works, the existence of an equilibrium is proved by showing that it corresponds to an optimal solution of an associated convex utility maximization problem. This, however, implies that every user possibly splits the flow among an exponential number of routes which might be critical for some applications. For instance, the standard TCP/IP protocol suite uses single path routing, because splitting the demand comes with several practical complications, e.g., packets arriving out of order, packet jitter due to different path delays etc. This issue has been explicitly addressed by Orda et al. [22].

Another related class of games are *congestion games*, where there is a set of resources, and the pure strategies of players are subsets of this set. Each resource has a delay function depending on the *load*, i.e., the number of players that select strategies containing the respective resource. These games allow to model network structures, but they fail to incorporate a realistic allocation of network capacities. The reason is that, even though we can define bandwidths allocated on an edge as a function of the number of players using it, the bandwidth of a player would be given by the *sum* of bandwidth allocated on edges he uses. This problem is addressed by *bottleneck congestion games* [6] where the bandwidth of one player is rather given by the *maximum* bandwidth among the edges he uses. It is known that strong equilibria exist for bottleneck congestion games [11]. The complexity of computing PNE and SE in these games was further investigated in [9], where a central result is an algorithm called Dual Greedy that computes SE. On single-commodity network or matroid bottleneck congestion games, it can be implemented to run in polynomial time. Still, for an arbitrary state, the computation of a coalitional improvement step turned out to be NP-hard, even for these classes. The PoA for PNE in bottleneck games can be polynomial in the network size, for social cost being the sum of player delays [6] or maximum player delay [5], [3].

A fundamental drawback of bottleneck congestion games is that the bandwidth allocated to a player on a specific edge is *solely* a function of the number of players on it. If one of the players does not exhaust his allocated bandwidth share (e.g., because he has a smaller bottleneck on another edge) the remaining bandwidth remains unused. In max-min fair allocations [12], this leftover is fairly distributed among players who can make use of it.

Yang et al. [29] introduced so-called *MAXBAR*-games which correspond to progressive filling games using max-min fair allocations. They show that these games possess PNE and that the price of anarchy for PNE is n in these games, where n is the number of players. It is also shown that iterative computation of unilateral improvement steps converge in polynomial time to a PNE if the number of players is constant.

In terms of combinatorial optimization, the problem of computing a strategy with maximum aggregated bandwidth (without fairness constraints) is related to the *maximum k-splittable flow problem* [2]. In contrast to the ordinary maximum flow problem, the number of paths flow is sent along is bounded by k, for each commodity. Positive results were especially found for the single-commodity case. For $k = 2$ and $k = 3$, a $\frac{3}{2}$-approximation was given and this result was generalized to a 2-approximation for arbitrary fixed k. It turned out that, asymptotically, any approximation with a factor of smaller than $\frac{6}{5}$ is NP-hard to obtain. Furthermore, for $k = 2$, $\frac{3}{2}$ is exactly the inapproximability bound [18].

II. Progressive Filling Game

A *progressive filling game* is a tuple $\left(N, R, (c_i)_{r\in R}, (\mathcal{S}_i)_{i\in N}, (v_i)_{i\in N}, (u_i)_{i\in N}\right)$, where $N = \{1, \ldots, n\}$ is the set of players, $R = \{1, \ldots, m\}$ is the set of resources, $c_r \in \mathbb{R}_+$ is the capacity of resource r for each $r \in R$. The allocation rate is defined as $v_i : \mathbb{R}_+ \to \mathbb{R}_+$ and is assumed to be (Riemann) integrable. The aggregated rate (or bandwidth) of player i at time t' is defined as $V_i(t') = \int_0^{t'} v_i(t)\, dt$. We assume that for all $i \in N$, $V_i \geq 0$, $V_i(t)$ is monotonically non-decreasing in t, and $\lim_{t\to\infty} V_i(t) = \infty$. We denote by $\mathcal{S}_i \subseteq \mathcal{P}(R)$ the set of strategies of player i, for each $i \in N$, and $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$ are the set of states. Note that this definition is kept very general and can be restricted to model more specific objects, e.g. networks. An *allocation* in state $S \in \mathcal{S}$ is a vector $a = (a_1, \ldots, a_n) \in \mathbb{R}^n$ of feasible bandwidths, i.e., $\sum_{i\in N: r\in S_i} a_i \leq c_r$, for each $r \in R$. The i-th component of a is called the bandwidth or capacity of player i (in a). Given S, we create an allocation the following way. Each of the players starts off with a bandwidth $b_i = 0$. We raise their bandwidths with the velocity $v_i(t)$ at time step $t \in \mathbb{R}$ until a further increase would lead to non-feasible capacities (i.e., one of the resources is *saturated*). At this point, we fix the bandwidths of all the corresponding (*saturated*) players and continue with the other ones. See Algorithm 1 for a formal description. For given S, we denote by $t_i(S)$ the *finishing time*, i.e., the time when player i's bandwidth is fixed. Thus, the payoff for player i is given by $u_i(S) = V_i(t_i(S)) = a_i$. We can easily extend our model to allow for player-specific payoff functions of the form $u_i(S) = U_i(V_i(t_i(S))) = U_i(a_i)$, where U_i is a differentiable and strictly increasing bandwidth utility function. As long as U_i is strictly increasing (yielding a monotone payoff transformation), an allocation is a PNE (SE) in the new game iff it is one in the original game. We now state a useful observation linking the outcome of Algorithm 1 with different fairness concepts.

Proposition 1. *Let $U_i, i \in N$ be a set of nonnegative, differentiable and strictly increasing bandwidth utility functions and let $w_i, i \in N$ be a set of nonnegative weights. For given progressive filling game and state S, the following holds:*

1) *If for all $i \in N : v_i(t) = 1$ and $u_i(S) = V_i(t_i(S))$, Algorithm 1 computes a max-min fair bandwidth allocation under S.*
2) *If for all $i \in N : v_i(t) = w_i$ and $u_i(S) = V_i(t_i(S))$, Algorithm 1 computes a weighted max-min fair bandwidth allocation under S.*
3) *If for all $i \in N : v_i(t) = \frac{d}{dt}\left(U_i^{-1}(t)\right)$ and $u_i(S) = U_i(V_i(t_i(S)))$, Algorithm 1 computes a utility max-min fair bandwidth allocation under S.*

Proof: As (1) and (2) are known in the literature (cf. [4]) we only prove (3). In order to obtain a utility max-min fair allocation, we need to ensure that while raising rates, the bandwidth utilities must be equally distributed. Thus, starting with $t = 0$ we set $U_i(V_i(t) = t$ for all $i \in N$. This is equivalent to $U_i^{-1}(t) = V_i(t)$ using that U_i is strictly increasing and thus invertible. Differentiating both sides leads to $v_i(t) = \frac{d}{dt}\left(U_i^{-1}(t)\right)$ as claimed. Since U_i is strictly increasing, its inverse is also strictly increasing (and also nonnegative), hence, $v_i(t)$ satisfies all assumptions needed. Now it follows by standard arguments (cf. [8]) that the resulting allocation is utility max-min fair. ■

Algorithm 1 Progressive Filling(PF)

Parameters: A progressive filling game $\mathcal{G} = \left(N, R, (c_i)_{r \in R}, (\mathcal{S})_{i \in N}, (v_i)_{i \in N}\right)$
Input: A state $S = (S_1, \ldots, S_n) \in \mathcal{S}$.
Output: The bandwidth b_i for each player $i \in N$.

$b_i \leftarrow 0$, **for all** $i \in N$; $N' \leftarrow N$
$N_r \leftarrow \{i \in N \mid r \in S_i\}$; $c'_r \leftarrow c_r$, **for all** $r \in R$
while $N' \neq \emptyset$ **do**
  $t^\star \leftarrow \min\{t' \mid \exists r \in R$ with $\sum_{i \in N_r \cap N'} \int_0^{t'} v_i(t)\,dt = c'_r$ and $N_r \cap N' \neq \emptyset\}$
  choose $r^\star$ **with** $\sum_{i \in N_{r^\star} \cap N'} \int_0^{t^\star} v_i(t)\,dt = c'_{r^\star}$ **and** $N_{r^\star} \cap N' \neq \emptyset$
  for each $i \in N_{r^\star} \cap N'$ **do**
    $b_i \leftarrow \int_0^{t^\star} v_i(t)\,dt$
    $N' \leftarrow N' \setminus \{i\}$
    for each $r \in S_i$ **do**
      $c'_r \leftarrow c'_r - b_i$
    end for
  end for
end while
return $(b_1, \ldots, b_n)$

III. Existence of Equilibria

We first study game-theoretic properties of a progressive filling game. We show that SE exist and moreover every sequence of improving deviations of coalitions converges to a SE.

Theorem 2. *Every progressive filling game has a SE, and every sequence of improving deviations of coalitions converges to a SE.*

Proof: Let $\mathcal{G}$ be a PFG and S a state in this game. For a player i, recall that we denote by $t_i(S)$ the finishing time, i.e., the point in time when his bandwidth is fixed by Algorithm 1 on S. Likewise, we denote by $\tilde{t}_r(S)$ the point in time when resource r gets saturated. In the remainder of the proof we crucially exploit the monotone relationship between the obtained bandwidth and the finishing time of every player. By the monotonicity of the V_i's, if a player strictly improves his obtained bandwidth by using an alternative strategy, then the new finishing time must strictly increase.

For a state S, we define a lexicographical potential function $\phi : \mathcal{S} \to \mathbb{R}_+^n$ as the vector of finishing times sorted in non-decreasing order, i.e., $\phi(S) = (t_{i_1}(S), \ldots, t_{i_n}(S))$ with $\{i_1, \ldots, i_n\} = N$ and $t_{i_j}(S) \leq t_{i_{j+1}}(S)$.

The next lemma shows that in a state S an improving move of a coalition C to a state T implies that $\phi(S) \prec \phi(T)$ where $\prec$ denotes the lexicographic ordering of vectors. Thus, a $\prec$-maximal state must be a SE. This implies the existence of the potential function and thereby the theorem. ∎

Lemma 3. *Let $C \subseteq N$ be a coalition which has an improving move from $S = (S_1, \ldots, S_n)$ to $T = (T_1, \ldots, T_n)$ where $S, T \in \mathcal{S}$. Then we have*

(a) $t_i(T) \geq t_i(S)$, for all $i \in N$ with $t_i(S) \leq \min_{j \in C} t_j(S)$, and
(b) $t_i(T) > \min_{j \in C} t_j(S)$, for all $i \in N$ with $t_i(S) > \min_{j \in C} t_j(S)$.

Proof: For some player i, note that we have $\tilde{t}_r(S) > t^\star$ for all $r \in T_i$ if and only if $t_i(S) > t^\star$ for some $t^\star \in \mathbb{R}$. Hence, it suffices to show that

(a′) $\tilde{t}_r(T) \geq \tilde{t}_r(S)$, for all $r \in R$ with $\tilde{t}_r(S) \leq \min_{j \in C} t_j(S)$, and
(b′) $\tilde{t}_r(T) > \min_{j \in C} t_j(S)$, for all $r \in R$ with $\tilde{t}_r(S) > \min_{j \in C} t_j(S)$.

For all $r \in T_i$ for some $i \in C$, the claim directly follows because we have $t_i(T) > t_i(S) \geq \min_{i \in C} t_i(S)$ and thus $\tilde{t}_r(T) > \min_{i \in C} t_i(S)$. So let $r \in R$ such that r is not used in T by any player from C.

In S, no resource which is used by a player from C has been saturated before $\min_{i \in C} t_i(S)$. Consequently, the bandwidth allocated to a player i is identical at time $\min_{i \in C} t_i(S)$ in S and T for all $i \in N \setminus C$. Since in T resource r is used by exactly the same players from $N \setminus C$ as in S and by no player from C, the residual capacity of r at a time $t \leq \min_{i \in C} t_i(S)$ is in T at least as high as in S.

This last result immediately implies (a′). For (b′), let $\tilde{t}_r(S) > \min_{j \in C} t_j(S)$. This means that the residual capacity at time $\min_{i \in C} t_i(S)$ is above zero in S and hence also in T. By the continuity of the indefinite integrals of the allocation rate functions, we obtain $\tilde{t}_r(T) > \min_{j \in C} t_j(S)$. ∎

Note that the above result applies to PFGs in full generality, that is, only requiring that the functions V are non-negative, non-decreasing, and tend to infinity for t going to infinity. We now show that the assumptions underlying this result cannot be relaxed. Clearly, relaxing non-negativity or relaxing the unboundedness of V makes not much sense. Negative aggregated rates have no physical meaning and for a bounded V there exists a game with large enough capacities for which Algorithm 1 does not terminate. More interestingly, suppose we have an allocation rate function for which the aggregated bandwidth $V(t)$ is non-monotonic. Note that this extension still allows to use Algorithm 1 to calculate the allocation via progressive filling. We show that for any such function, Theorem 2 does not hold anymore. This is even true if we restrict to two-player games with symmetric strategy spaces.

Theorem 4. *Let v be such that $V : \mathbb{R}_+ \to \mathbb{R}_+, t' \mapsto \int_0^{t'} v(t)\,dt$ satisfies $V \geq 0$ and $\lim_{t \to \infty} V(t) = \infty$. If $V(t)$ is not monotone, there is a two-player PFG $\mathcal{G}_v$ with symmetric strategy spaces that does not have a PNE and only uses v and one constant function as allocation rate functions.*

Proof: Let v be an allocation rate function such that the aggregated rate function V is not monotone. By the continuity and non-negativity of V, there is $t_1 > 0$ such that for every $\epsilon > 0$, there is $t_2 = t_2(\epsilon) \in (t_1, t_1 + \epsilon)$ with $V(t_1) > V(t_2)$ (see [10, Lemma 3.1]). Thus, we can choose t_2 satisfying $t_2 < t_1 + \epsilon$ for any $\epsilon > 0$ to be specified later. Since v is Riemann integrable and thus on the interval $[0, t_2]$ bounded, its indefinite integral V has a Lipschitz constant $\rho > 0$ on $[0, t_2]$.

We now describe the game $\mathcal{G}_v$ with two players $\{1, 2\}$. We set $R = \{r_1, r_2, r_3\}$ with $c_{r_1} = c_{r_2} = (\rho + 1)t_1 + V(t_1)$ and $c_3 = (\rho + 1)t_2 + V(t_2)$. Furthermore, the sets of strategies are

$\mathcal{S}_1 = \mathcal{S}_2 = \{\{r_1, r_3\}, \{r_2, r_3\}\}$. As allocation rate functions, we use $v_1 \equiv v$ and $v_2 \equiv \rho + 1$. We claim that, whenever both players share one of the resources r_1 or r_2, the shared resource is saturated at time t_1 and player 2 gets bandwidth $(\rho+1)t_1$ while player 1 gets bandwidth $V(t_1)$. To see this, we use the Lipschitz inequality $\frac{V(t)-V(t_1)}{t_1-t} < (\rho+1)$ for all $t \in [0, t_1)$ implying $(\rho+1)t + V(t) < (\rho+1)t_1 + V(t_1)$ for all $t \in [0, t_1)$. On the other hand, whenever player 2 is alone on either r_1 or r_2, resource r_3 is saturated at time t_2 using again $\frac{V(t)-V(t_2)}{t_2-t} < (\rho+1)$ for all $t \in [0, t_2)$. By choosing $t_2 < t_1 + V(t_1)/(\rho+1)$ (hence $t_2 = t_2(\epsilon)$ with $\epsilon = V(t_1)/(\rho+1)$) we get $(\rho+1)t_2 < (\rho+1)t_1 + V(t_1)$ and, thus, none of the resources r_1 or r_2 gets saturated before t_2. Consequently, player 2 gets bandwidth $(\rho+1)t_2 > (\rho+1)t_1$ while player 1 gets bandwidth $V(t_2) < V(t_1)$. Hence, there is no PNE. ■

IV. Max-Min-Fair Progressive Filling Games

A special case of progressive filling games arises if all players raise their bandwidth uniformly, i.e., $v_i(t) = 1$ for all $i \in N$. This leads to allocations that are max-min fair. We call such a game max-min-fair progressive filling game or MMFG. More formally, let $S \in \mathcal{S}$ be a state and $\mathcal{A} = \{a \mid a \text{ is an allocation in } S\}$, then the unique $\preceq$-maximal $a^\star$ in $\mathcal{A}$ is the max-min fair allocation. In the following, we will study the computational complexity and efficiency of SE and PE in MMFGs.

A. Computing Equilibria

Similar to [9], we use a *dual greedy algorithm* [23] to compute strong equilibria. Our dual greedy algorithm is allowed to query a *strategy packing oracle* that solves the strategy packing problem which is the following: The input is given by a set R of $m \in \mathbb{N}$ resources, n sets of strategies $\mathcal{S}_i \in \mathcal{P}(R)$, for all $i \in \{1, \ldots, n\}$, along with upper bounds $u_r \in \{0, \ldots, n\}$, for each $r \in R$. The output is a state $(S_1, \ldots, S_n) \in \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$ satisfying the upper bounds, i.e., $|\{i \in N \mid r \in S_i\}| \le u_r$, for all $r \in R$, if it exists. Otherwise the output is the information that no such state exists.

The dual greedy algorithm initially allows an upper bound of $u_r = n$ players on each resource r and every resource and every player is initially considered *free*. The algorithm starts with an arbitrary state S of strategies for players. It iteratively decrements one of the bounds u_r on a free resource providing minimum bandwidth if each resource was used by u_r players. After each decrement, it checks the existence of a strategy profile respecting the new upper bounds on the number of players using it by invoking the strategy packing oracle. When a decrease produces infeasible bounds, i.e., when there is no state of the game respecting the new bounds, it reverts the last decrease. Now we know that in the profile that was returned by the oracle, exactly u_r players are using r and it is infeasible to further reduce u_r. Thus, the algorithm turns r into a *fixed* resource, and also fixes the u_r players as well as their strategies. In addition, it decreases every resource capacity by the amount given to the u_r fixed players in their strategies. Then it continues with the remaining players, resources, and residual capacities. For a formal statement of the algorithm see Algorithm 2.

Theorem 5. *The dual greedy algorithm computes a SE.*

Algorithm 2 Dual Greedy Algorithm

Let $\mathfrak{O}$ denote the strategy packing oracle.
Input: A MMFG $\mathcal{G} = \left(N, R, (c_i)_{r \in R}, (\mathcal{S})_{i \in N}\right)$
Output: A SE in $\mathcal{G}$.

$b_i \leftarrow 0$, **for all** $i \in N$; $N' \leftarrow N$
$u_r \leftarrow n, c'_r \leftarrow c_r$, **for all** $r \in R$
while $N' \neq \emptyset$ **do**
  $(S'_i)_{i \in N'} \leftarrow \mathfrak{O}\left(R, (\mathcal{S}_i)_{i \in N'}, (u_r)_{r \in R}\right)$
  choose $r^\star \in \arg\min_{r \in E: u_r > 0} \frac{c'_r}{u_r}$
  $u_{r^\star} \leftarrow u_{r^\star} - 1$
  if $\mathfrak{O}\left(R, (\mathcal{S}_i)_{i \in N'}, (u_r)_{r \in R}\right) = \emptyset$ **then**
    $u_{r^\star} \leftarrow u_{r^\star} + 1$
    $b \leftarrow \frac{c'_{r^\star}}{u_{r^\star}}$
    for each $i \in N'$ **with** $r^\star \in S'_i$ **do**
      $S_i \leftarrow S'_i$
      $N' \leftarrow N' \setminus \{i\}$
      for each $r \in S_i$ **do**
        $u_r \leftarrow u_r - 1$
        $c'_r \leftarrow c'_r - b$
      end for
    end for
  end if
end while
return S

The main idea of the proof (contained in the full version) is similar to [9], i.e., the iterative assignment of Dual Greedy yields a lexicographically maximal vector of bandwidths. Consider on each resource the residual capacity not yet assigned to fixed players. We can assume that this residual capacity is offered in equal shares to the remaining free players. Thus, the share of each free player only depends on the number of free players using it. Hence, as long as no players are fixed, the game can be seen equivalently as a bottleneck congestion game. In addition, once a resource and players are fixed, then the bandwidth of a fixed player is smaller than the equal share of residual capacity on every free resource he uses. This allows to inductively show correctness of the algorithm.

Dual Greedy can be implemented in polynomial time given an efficient strategy packing oracle. Hence, the problem of computing SE in MMFGs is polynomial-time reducible to the strategy packing problem. There are several non-trivial cases in which the strategy packing problem is polynomial-time solvable, e.g., for single-commodity networks [9]. Thus, we obtain the following result.

Corollary 6. *SE can be computed in polynomial time for single-commodity network MMFGs.*

In contrast, the strategy packing problem turns out to be NP-hard even if we generalize to symmetric strategy spaces. This result permits computation of SE polynomial time by other algorithms than Dual Greedy, but, in fact, computation of SE and strategy packing are *mutually* polynomial-time reducible, even for symmetric games.

Theorem 7. *The strategy packing problem for symmetric strategies is NP-hard. Moreover, the computation of a SE in symmetric MMFGs is NP-hard.*

B. Efficiency of Equilibria

In this section we investigate the quality of SE in terms of social welfare, i.e., the sum of allocated bandwidth. In a game $\mathcal{G}$, let $S^\star$ with allocation a be the state in $\mathcal{S}$ that maximizes $\sum_{i \in N} a_i$. Further, let $\mathcal{S}^{SE} \subseteq \mathcal{S}^{NE} \subseteq \mathcal{S}$ denote the set of SE and NE, respectively. We denote $\mathrm{SW}_{\mathcal{G}}(S) = \sum_{i \in N} b_i(S)$. Then, the price of stability and price of anarchy, PoS and PoA, are defined as $\inf_{S \in \mathcal{S}^{NE}} \frac{\mathrm{SW}_{\mathcal{G}}(S^\star)}{\mathrm{SW}_{\mathcal{G}}(S)} = \inf_{S \in \mathcal{S}^{NE}} \frac{\sum_{i \in N} a_i}{\sum_{i \in N} b_i(S)}$ and $\sup_{S \in \mathcal{S}^{NE}} \frac{\mathrm{SW}_{\mathcal{G}}(S^\star)}{\mathrm{SW}_{\mathcal{G}}(S)}$, respectively. For the strong price of stability and anarchy, SPoS and SPoA, $\mathcal{S}^{SE}$ is considered instead of $\mathcal{S}^{NE}$. Furthermore, the same measures can be applied to classes of games where they are simply the supremum of all individual measures.

The *maximum capacity allocation problem (MCAP)* is given by the problem of computing an allocation a' which maximizes $\sum_{i \in N} a'_i$. Note that we have $\sum_{i \in N} a'_i \geq \sum_{i \in N} a_i$ and that this inequality may even be strict since a' is not necessarily computed by progressive filling.

In general, one cannot hope to find SE with good social welfare. There are network MMFGs in which even the best PNE is a factor of $\Omega(n)$ worse than the optimum. This matches the upper bound of $O(n)$ on the PoA for network MMFGs shown in [29].

Theorem 8. *The PoS and SPoS in multi-commodity network MMFGs are $\Omega(n)$.*

Proof: For a given $n \in \mathbb{N}$, we construct a network MMFG $\mathcal{G}_n$ with n players and PoS of more than $\frac{n}{4}$.We assume w.l.o.g. that $2 \mid n$.

The network underlying $\mathcal{G}_n$ consists of $\frac{n}{2}$ consecutive edges each of which has capacity 1 and connects the source and sink nodes s_i, t_i of one respective player i. The source and sink nodes of the other $\frac{n}{2}$ players are the first and last vertex of this path. Additionally, there is one edge with the capacity 0 between these two vertices. This network is illustrated in Figure 1.

At first, note that there are two strategies for each of the players. A player can either choose the path through the 1-edges or the path which has the 0-edge in it. The latter path will, however, not be taken in a PNE since avoiding the 0-edge always results in a bandwidth strictly larger than 0. Thus, in the unique PNE S, each 1-edge is congested with $\frac{n}{2}+1$ players, resulting in a social welfare of $\mathrm{SW}_{\mathcal{G}_n}(S) = n \cdot \frac{1}{\frac{n}{2}+1} = \frac{2n}{n+2}$.

If, however, the players $i \in \left\{\frac{n}{2}+1, \ldots, n\right\}$ altruistically take the direct 0-capacity path (s_i, t_i) instead, all the other players get a bandwidth of 1 by sticking to their paths from S. Consequently, a lower bound on the PoS is $\frac{\frac{n}{2}}{\mathrm{SW}_{\mathcal{G}_n}(S)} = \frac{\frac{n}{2}}{\frac{2n}{n+2}} = \frac{n+2}{4} > \frac{n}{4}$. ■

In contrast, when all players have the same strategy set, the best SE achieves a good approximation, and such a good SE is found by Dual Greedy (for single-commodity networks even in polynomial time).

Theorem 9. *The PoS and SPoS in symmetric MMFGs are $2 - \frac{1}{n}$, and this bound is tight. The Dual Greedy computes an SE achieving this guarantee.*

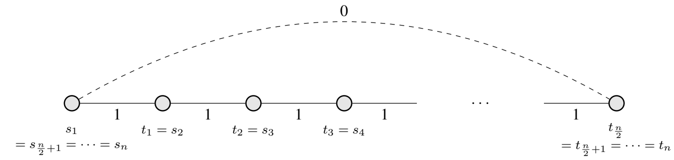

Fig. 1: Illustration of the network of the game $\mathcal{G}_n$ in the proof of Theorem 8.

Proof: For the upper bound, we use an idea from [2] and define the *uniform MCAP* as the restriction of the MCAP to uniform bandwidth values, i.e., we additionally require that the found allocation is a vector $(a, \ldots, a)$ for some $a \in \mathbb{R}$. It is easy to see that the smallest bandwidth in the state S_{DG} computed by Dual Greedy solves the uniform MCAP. That is $\min_{i \in N} b_i(S_{DG}) = v$ where $n \cdot v$ is the optimal value of the uniform MCAP (see a corresponding Lemma in the full version). Thus, the upper bound follows from the next lemma.

Lemma 10. *An optimum to the uniform MCAP is a $\left(2 - \frac{1}{n}\right)$-approximation for the MCAP.*

For the lower bound consider for given $n \in \mathbb{N}$ and $\varepsilon \in \mathbb{R}$ with $\varepsilon > 0$ a single-commodity network MMFG $\mathcal{G}_{n,\varepsilon}$. From the source to the sink node, there are $n-1$ parallel edges each of which has capacity $1-\varepsilon$. Moreover, there is one single edge with capacity n. In the optimal state, every edge is used by one player each, i.e., we obtain a social welfare of $2n-1-(n-1)\cdot\varepsilon$. In a NE, however, every player uses the edge with capacity n because the bandwidth for each player is at least $1 > 1-\varepsilon$ on this edge. Thus, the social welfare is exactly n in this state. Therefore, it holds that $\mathrm{PoS}(\mathcal{G}_{n,\varepsilon}) = \frac{2n-1-(n-1)\cdot\varepsilon}{n} = 2 - \frac{1}{n} - \frac{n-1}{n} \cdot \varepsilon$. ■

In symmetric games even the worst SE is still a 4-approximation. For $n = 2$, we can tighten this to the SPoS-bound of $\frac{3}{2}$. In the full version we also show a lower bound of $\Omega(n/k)$ on the k-SPoA for k-SE, where only deviations of coalitions of size at most k are considered.

Theorem 11. *The SPoA for symmetric MMFGs is at most $4 - \frac{6}{n+1}$.*

Proof: Let $\mathcal{G}$ be a symmetric MMFG and let S be a SE in this game. Then in S each player must get at least a bandwidth of $\frac{1}{2}\max_{i \in N} b_i(S)$, as otherwise this player could profitably imitate a player in $\operatorname{argmax}_{i \in N} b_i(S)$ by choosing the same strategy. Thus, we can lower bound the social welfare by

$$\mathrm{SW}_{\mathcal{G}}(S) = \sum_{i \in N} b_i(S) \geq \left(\frac{n-1}{2} + 1\right) \cdot \max_{i \in N} b_i(S). \quad (1)$$

State S_{DG} computed by Dual Greedy in $\mathcal{G}$ is such that $\min_{i \in N} b_i(S_{DG}) = v$ where $n \cdot v$ is the optimal value of the uniform MCAP. Consequently, for any other SE S, we must have $\max_{i \in N} b_i(S) \geq v$, because otherwise all the players could profitably switch to their strategies in S_{DG}. Using Equation 1, this means $\mathrm{SW}_{\mathcal{G}}(S) \geq \frac{n+1}{2} \cdot v$, and hence we obtain

$$\frac{\max_{S'\in\mathcal{S}} \mathrm{SW}_{\mathcal{G}}(S')}{\mathrm{SW}_{\mathcal{G}}(S)} \leq \frac{2n}{n+1} \cdot \frac{\max_{S'\in\mathcal{S}} \mathrm{SW}_{\mathcal{G}}(S')}{n \cdot v} \leq \frac{2n}{n+1} \cdot \frac{2n-1}{n} = \frac{4n-2}{n+1}.$$

■

Theorem 12. *The SPoA for symmetric MMFGs with 2 players is $\frac{3}{2}$.*

Proof: Let $\mathcal{G}$ be a symmetric MMFG with $n = 2$ and let S be a SE in this game. Further, let S' be an arbitrary (optimal) state. W.l.o.g., we may assume that $b_1(S) \leq b_2(S)$ and $b_1(S') \leq b_2(S')$.

Note that $b_1(S) \geq b_1(S')$ or $b_2(S) \geq b_2(S')$ must hold. Otherwise switching from S to their strategies in S' would be profitable for both players. Thus, the following case distinction is complete.

Case 1: We have $b_1(S) \geq b_1(S')$. We can also derive an upper bound on $b_2(S')$. If $b_2(S') > 2 \cdot b_1(S)$, player 1 could profitably deviate to S'_2 in S. So we must have $b_2(S') \leq 2 \cdot b_1(S)$. Thus,

$$\frac{\mathrm{SW}_{\mathcal{G}}(S')}{\mathrm{SW}_{\mathcal{G}}(S)} = \frac{b_1(S') + b_2(S')}{b_1(S) + b_2(S)} \leq \frac{3 \cdot b_1(S')}{2 \cdot b_1(S)} \leq \frac{3}{2}.$$

Case 2: We have $b_2(S) \geq b_2(S')$. We find an upper bound on $b_2(S)$. Since $b_1(S) < \frac{1}{2} \cdot b_2(S)$ would mean that player 1 could profitably imitate player 2 in S, it holds that $b_1(S) \geq \frac{1}{2} \cdot b_2(S)$. This implies

$$\frac{\mathrm{SW}_{\mathcal{G}}(S')}{\mathrm{SW}_{\mathcal{G}}(S)} = \frac{b_1(S') + b_2(S')}{b_1(S) + b_2(S)} \leq \frac{2 \cdot b_2(S')}{\frac{3}{2} \cdot b_2(S)} \leq \frac{4}{3} < \frac{3}{2}.$$

The lower bound immediately follows from Theorem 9. ■

V. General Progressive Filling Games

A. Complexity and Convergence

The lexicographical potential function for PFGs implies that the length of each coalitional improvement sequence is finite. By Φ_{ord}, we denote the set of ordered values of ϕ, i.e., $\Phi_{\mathrm{ord}} := \mathrm{img}(\mathrm{ord} \circ \phi)$ where ord is the function which orders a vector, say ascendingly. The cardinality of the above set provides an upper bound on the length of improvement sequences.

For a MMFG with n players and m resources, Yang et al. [29] provide an upper bound of $(mn)^n$ on the number of improvement steps to reach a PNE. In the following, we show that it is not possible to get this result by just bounding $|\Phi_{\mathrm{ord}}|$.

Theorem 13. *There is a family of network MMFGs $\mathcal{G}_n$ with $m \in \Theta(n)$ and respective potential function Φ_{ord} such that $|\Phi_{\mathrm{ord}}|$ is in $2^{\Omega(n^2)} = \omega\left((n^2)^n\right)$.*

Proof: Since we are proving an asymptotical lower bound, we may assume w.l.o.g. that $2 \mid n$. We now describe the multigraph underlying $\mathcal{G}_n$.

For each player $i \in \left\{\frac{n}{2}+1, \ldots, n\right\}$, we have a gadget in this multigraph. This gadget consists of two parallel edges, both connecting the source and the sink nodes (s_i and t_i, respectively) of the particular player. One of these edges (referred to as the *left* edge) has a capacity of $2^{\frac{n}{2}+1}$ whereas the other one also has at least this capacity. The gadgets are arranged in a row such that $s_i = t_{i+1}$ for $i \in \left\{\frac{n}{2}+1, \ldots, n-1\right\}$. All the other players $i \in \left\{1, \ldots, \frac{n}{2}\right\}$ have one disjoint source node s_i each and t_n as sink node. Moreover, there is one edge connecting s_i and $s_{\frac{n}{2}}$ with capacity 2^{i-1}. This results in a network as shown in Figure 2. Obviously, the number of edges is in $\Theta(n)$.

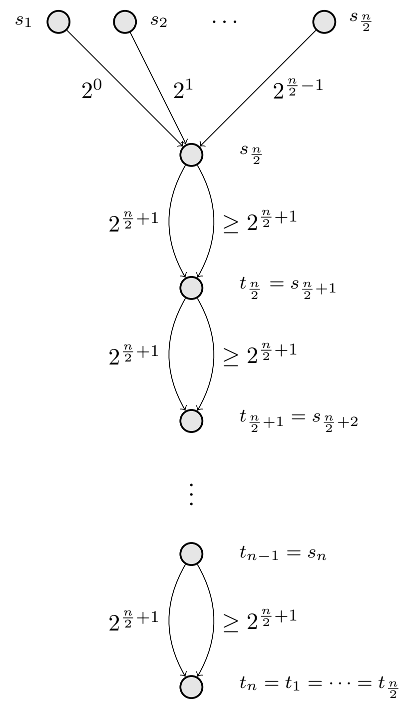

Fig. 2: Illustration of the network in the proof of Theorem 13.

Note that, independently of the path a player $i \in \left\{1, \ldots, \frac{n}{2}\right\}$ chooses, he is always assigned the respective bandwidth 2^{i-1} in the max-min fair allocation. This is because the residual capacity of an edge from the gadgets is larger than each of the bandwidth of players from $\left\{1, \ldots, \frac{n}{2}\right\}$, even if all these players use this edge.

Consequently, the players $\left\{1, \ldots, \frac{n}{2}\right\}$ are capable of choosing any natural number between $2^{\frac{n}{2}}+1$ and $2^{\frac{n}{2}+1}$ for the residual capacity of each of the $\frac{n}{2}$ left edges in the different gadgets. More specifically, let $x_{\frac{n}{2}} x_{\frac{n}{2}-1} \ldots x_1$ be the binary representation of a natural number x such that $2^{\frac{n}{2}+1} - x$ is from that interval. To obtain the desired residual capacity on a left edge in a given gadget, player i simply chooses this edge in his path if and only if we have $x_i = 1$. This has indeed the desired effect since $2^{\frac{n}{2}+1} - \sum_{i\in N: x_i=1} 2^{i-1} = 2^{\frac{n}{2}+1} - x$. We now give a lower bound on the number of different ordered allocation vectors. Since we want to derive a lower bound, it suffices to show the claim for allocations where the residual capacity of the left edge in the i-th gadget (i.e. the one of player $\frac{n}{2}+i$) is between $2^{\frac{n}{2}}+1+(i-1)\cdot\left\lfloor(2^{\frac{n}{2}}-1)/n\right\rfloor$ and $2^{\frac{n}{2}}+1+i\cdot\left\lfloor(2^{\frac{n}{2}}-1)/n\right\rfloor$ and player i chooses this edge. In these allocations, the bandwidth of player i occurs in the ordered allocation vector *before* the one of player $i+1$, for all $i \in \left\{\frac{n}{2}+1,\ldots,n-1\right\}$. Consequently, the claim is implied by the following bound on the number of ordered allocations

$$\left\lfloor\frac{2^{\frac{n}{2}}-1}{n}\right\rfloor^{\frac{n}{2}} = \left(\frac{2^{\Omega(n)}}{2^{\mathcal{O}(\log n)}}\right)^{\Omega(n)} = 2^{\Omega(n)\cdot\Omega(n)} = 2^{\Omega\left(n^2\right)}.$$

■

We now provide an *upper* bound on the number of ordered values of the potential, even for general progressive filling games. For $m = \Theta(n)$, this yields an upper bound of $2^{O(n^2)}$.

Theorem 14. *For arbitrary PFGs with the potential function ϕ, it holds that $|\Phi_{\text{ord}}| \leq 2^{n^2} \cdot m^n$.*

Proof: Let $\mathcal{G}$ be a PFG with potential function ϕ. We claim that the number of different vectors up to the k-th position (for $k \leq n$) in $|\Phi_{\text{ord}}|$ is at most $2^{k\cdot n} \cdot m^k$. It is shown via induction on k.

For $k = 0$, the claim is clear as there is only the vector of dimension 0. So let $n \geq k > 0$ and assume there are at most $2^{(k-1)\cdot n} \cdot m^{(k-1)}$ different vectors up to the position $k-1$ in Φ_{ord}. We now fix the first $k-1$ positions of a vector in Φ_{ord} and bound the number of entries at the k-th position. Note that one can calculate the next finishing time given the resource which is saturated and the subset of players on that resource. Since there are $2^n \cdot m$ such combinations, the claim follows. ■

Theorem 7 shows that computing SE is NP-hard in MMFGs. For general PFGs with constant allocation rates (i.e., weighted MMF allocations), the same result holds even for single-commodity network games with two players. Hence, extending Dual Greedy to compute SE in polynomial time for this case is impossible.

Theorem 15. *Let $v_1 \neq v_2$ be two constant allocation rate functions and consider the class of single-commodity network PFGs with two players and v_1, v_2 as allocation rate functions. In this class, the computation of SE is NP-hard.*

Let us instead consider PNE, which may be easier to compute than SE. Similar to a result from [29] for MMFGs, we first show that one can efficiently compute a unilateral improvement step for a given player in a PFG with constant allocation rate functions (if it exists). Using Theorem 14, computation of PNE can be done efficiently for a constant number of players.

Lemma 16. *In PFGs with constant allocation rate functions, an improving move of any player i in any state S can be computed in polynomial time if it exists.*

Proof: The bandwidth of player i in the state $(\{r\}, S_{-i})$ can by computed in polynomial time by Algorithm 1. Further, for a given strategy S_i', the bandwidth of player 1 only depends on the resource which gets saturated first, i.e., $b_i(S_i', S_{-i}) = \min_{r\in S_i'} b_i(\{r\}, S_{-i})$, which can easily be verified on Algorithm 1. Thus, it suffices to calculate $\min_{r\in S_i'} b_i(\{r\}, S_{-i})$ for all possible alternative strategies S_i' to decide whether there is an improvement step from S for player i.

If the strategies are given explicitly as input, this value can be explicitly computed for each of the strategies. If strategies are given implicitly in the form of a network, we can use, e.g., Dijkstra's algorithm to find a path $P^\star$ with the maximum $\min_{r\in P^\star} b_i(\{r\}, S_{-i})$. ■

Corollary 17. *A PNE can be computed in polynomial time in PFGs with constant allocation rate functions and a constant number of players.*

B. Changing the Allocation Rate Functions

Dual Greedy computes a SE that is a $2-\frac{1}{n}$-approximation and this bound is tight. To stabilize better solutions, in this section we take a "protocol design" approach. We assume the waterfilling algorithm can determine a set of constant allocation rate functions for each instance. Interestingly, for any given collection of players, resources, capacities and strategy sets, one can give constant allocation rate functions such that the resulting PFG has an SE with social welfare as high as the optimal value of the MCAP.

Theorem 18. *Let $\mathcal{G}$ be a PFG with player set N and $v^\star$ be the optimal value of the MCAP. There are constant allocation rate functions $(v_i')_{i\in N}$ such that the maximal social welfare in $\mathcal{G}$ with allocation rate functions replaced by $(v_i')_{i\in N}$ is $v^\star$ and the SPoS in this game is 1.*

Proof: Let the state $S = (S_1,\ldots,S_n)$ along with the allocation $a = (a_1,\ldots,a_n)$ be an optimal solution of the MCAP. We use allocation rate function $v_i' \equiv a_i$ for each player $i \in N$. We call the corresponding PFG $\mathcal{G}'$. If we run the progressive filling algorithm in S with v', all finishing times are exactly 1 and the allocation is exactly a.
We show that S is a SE in $\mathcal{G}$. Towards this, suppose that there is a coalition profitably deviating from S to T. Then, by Lemma 3, the finishing times and thus bandwidths of all players in $N \setminus C$ remain identical in T whereas the players from C strictly improve. Consequently, we have constructed a solution of the MCAP on $\mathcal{M}$ with a higher social welfare – a contradiction. ■

Not surprisingly, this approach is intractable, as the MCAP is NP-hard to approximate to within a factor $\frac{3}{2}-\varepsilon$, even for arbitrary fixed rates (see the full version).

This implies that the approximation guarantee of Dual Greedy is optimal for $n = 2$, even without requiring the output to be a SE. The idea behind the previous theorem extends also to approximate solutions of the MCAP. For the MCAP on single-commodity networks, a better $\frac{3}{2}$-approximation exists for $n = 3$ [2] and can be obtained as follows: Run the maximum capacity augmenting path algorithm [1] on the given network for two iterations and decompose [1] the obtained flow into three paths (plus a circulation). We use this approach to calculate an *equilibrium state* that is a better approximation than the one calculated by Dual Greedy. By Theorem 9, this is not possible if the allocation rate functions are fixed, even for uniform ones. Adjusting allocation rate functions subject to the instance, however, allows to beat Dual Greedy, at least for $n = 3$ and PNE.

Theorem 19. *In single-commodity networks with 3 players, there exist constant allocation rate functions and a PNE that is a $\frac{3}{2}$-approximation to the MCAP. The allocation rate functions and the PNE can be computed in polynomial time.*

Indeed, we can start with an arbitrary approximate solution of the MCAP, set the allocation rates such that finishing times are all 1, and then every unilateral (coalitional) improvement dynamics will lead to a PNE (SE) that only improves social welfare. Exploring this idea is a very interesting avenue for future work.

ACKNOWLEDGEMENT

We thank Berthold Vöcking for helpful comments regarding the model underlying this paper.

REFERENCES

[1] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[2] G. Baier, E. Köhler, and M. Skutella. The k-splittable flow problem. *Algorithmica*, 42(3–4):231–248, 2005.

[3] R. Banner and A. Orda. Bottleneck routing games in communication networks. *IEEE J. Sel. Areas Commun.*, 25(6):1173–1179, 2007.

[4] D. Bertsekas and R. Gallager. *Data networks (2nd ed.)*. Prentice Hall, 1992.

[5] C. Busch and M. Magdon-Ismail. Atomic routing games on maximum congestion. *Theoret. Comput. Sci.*, 410(36):3337–3975, 2009.

[6] R. Cole, Y. Dodis, and T. Roughgarden. Bottleneck links, variable demand, and the tragedy of the commons. *Networks*, 60(3):194–203, 2012.

[7] H. Han, S. Shakkottai, C.V. Hollot, R. Srikant, D. Towsley. Multi-Path TCP: A Joint Congestion Control and Routing Scheme to Exploit Path Diversity in the Internet. *IEEE/ACM Trans. Networking*, 1(1): 22–33, 2006

[8] T. Harks and T. Poschwatta. Congestion Control in Utility Fair Networks. *Computer Networks*, 52(15): 2947–2960, 2008

[9] T. Harks, M. Hoefer, M. Klimm, and A. Skopalik. Computing pure Nash and strong equilibria in bottleneck congestion games. In *Proc. 18th ESA*, vol. 2, pages 29–38, 2010.

[10] T. Harks and M. Klimm. On the existence of pure Nash equilibria in weighted congestion games. *Math. Oper. Res.*, 37(3):419–436, 2012.

[11] T. Harks, M. Klimm, and R. Möhring. Strong Nash equilibria in games with the lexicographical improvement property. In *Proc. 5th WINE*, pages 463–470, 2009.

[12] J. Jaffe. Bottleneck flow control. *IEEE Trans. Commun.*, 29(7):954–962, 1981.

[13] F. P. Kelly and T. Voice Stability of end-to-end algorithms for joint routing and rate control *Comp. Comm. Rev.*, 35(2), 5–12, 2005.

[14] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate Control in Communication Networks: Shadow Prices, Proportional Fairness, and Stability. *J. Oper. Res. Soc.*, 49:237–52, 1998.

[15] P. Key, L. Massoulié, D. Towsley. Path selection and multipath congestion control. *Proc. 26th INFOCOM*, pages 143–151, 2007.

[16] S. Low, F. Paganini, and J. C. Doyle. Internet Congestion Control. *IEEE Control Systems Magazine*, 22, pp. 28–43, 2002.

[17] S. H. Low and D. E. Lapsley. Optimization Flow Control I. *IEEE/ACM Trans. Netw.*, 7(6):861–874, 1999.

[18] R. Koch and I. Spenke. Complexity and approximability of k-splittable flows. *Theoret. Comput. Sci.*, 369(1–3):338–347, 2006.

[19] L. Mamatas, T. Harks, and V. Tsaoussidis. Approaches to Congestion Control in Packet Networks. *J. Internet Engineering*, 1(1): 22–33, 2007

[20] K. Miller and T. Harks. Utility max-min fair congestion control with time-varying delays. *Proc. 27th INFOCOM*, pages 331–335, 2008.

[21] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Trans. Netw.*, 8(5), 556 – 567, 2000.

[22] Orda, A., R. Rom, N. Shimkin. Competitive routing in multi-user communication networks. *IEEE/ACM Trans. Networking*: 1, pp. 510–521, 1993.

[23] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer Verlag, 2003.

[24] R. Srikant. *The Mathematics of Internet Congestion Control*. Birkhaeuser, 2003.

[25] L. Tan, L. Dong, C. Yuan, and M. Zukerman. Fairness Comparison of FAST TCP and TCP Reno. *Comput. Commun.*, 30(6):1375–1382, 2007.

[26] J. Wang, L. Li, S. H. Low, and J. C. Doyle Cross-Layer Optimization in TCP/IP networks *IEEE/ACM Trans. Netw.*, 13(3):582–568, 2005

[27] W.-H. Wang, M. Palaniswami, S.H. Low. Optimal flow control and routing in multi-path networks. *Perform. Eval.*: 52: 119–132, 2003.

[28] B. Wydrowski and M. Zukerman. MaxNet: A congestion control architecture for MaxMin fairness. *IEEE Commun. Lett.*, 6, 512–514, 2002.

[29] D. Yang, G. Xue, X. Fang, S. Misra, and J. Zhang. Routing in max-min fair networks: A game theoretic approach. In *Proc. 18th ICNP*, pp. 1–10, 2010.

[30] Y. Zhang, S.-R. Kang, and D. Loguinov. Delay-independent stability and performance of distributed congestion control. *IEEE/ACM Trans. Netw.*, 15(4): 838–851, 2007.