Brief Announcement: Resource Allocation Games with Multiple Resource Classes

Roy B. Ofer and Tami Tamir(✉)

School of Computer Science, The Interdisciplinary Center, Herzliya, Israel
tami@idc.ac.il

1 Introduction

Media streaming is among the most popular services provided over the Internet. The lack of a central authority that controls the users, motivates the analysis of Media on Demand (MoD) services using game theoretic concepts. We define and study the corresponding resource-allocation game, where users correspond to self-interested players who choose a MoD server with the objective of minimizing their individual cost. Each user requires a certain media-file which determines the user's class. A server provides both broadcasting and storage needs. Accordingly, the user's cost function encompasses both negative and positive, class-dependent, congestion effects.

An instance of the multi-class resource allocation game is defined by a tuple $G = \langle I, M, A, U \rangle$, where I is the set of players, M is the set of servers and A is the set of classes. Let $n = |I|$ and $m = |M|$. Each player belongs to a single class from A, thus, $I = I_1 \cup I_2 \cdots \cup I_{|A|}$, where all players from I_k belong to class k. For $i \in I$, let $a_i \in A$ denote the class to which player i belongs. The parameter $U \in \mathbb{R}^+$ is the class activation-cost, which is assumed to be uniform for all classes.

An *allocation* of players to servers is a function $f : I \to M$. For a given allocation, the *load* on a server j, denoted by L_j, is the number of players assigned to j, and $L_{j,k}$ denotes the number of players from I_k assigned to j.

The cost of a player i in an allocation f consists of two components: the load on the server the player is assigned to (as in job scheduling games [3]), and the player's share in the class activation-cost (as in cost-sharing games [1]). Formally, $c_f(i) = L_{f(i)} + \frac{U}{L_{f(i),a_i}}$. Note that the class activation-cost is shared evenly among the players from this class serviced by a server. Our model generalizes the one studied in [2], where all players belong to the same class.

In MoD systems, the bandwidth required for transmitting a certain media-file corresponds to one unit of load. The storage cost of a media-file on a server is shared by the users requiring its transmission that are serviced by the server.

2 Our Results and Techniques

We provide answers to the basic questions regarding resource allocation games with multiple resource classes. Namely, equilibrium existence, convergence, calculation and efficiency. We prove that a *Pure Nash Equilibrium* (PNE) exists

M. Hoefer (Ed.): SAGT 2015, LNCS 9347, pp. 307–308, 2015.
DOI: 10.1007/978-3-662-48433-3_27

for any instance by presenting an exact potential function for the game. By analyzing this function we show:

Theorem 1. *For every instance G, better-response dynamics converges to a PNE within $O(n^4)$ steps.*

The equilibrium inefficiency is analyzed with respect to the objective of minimizing the maximal cost among the players. That is, given an allocation f, the social cost of f is given by $c_{max}(f) = \max_{i \in I} c_f(i)$.

We provide several lower bounds on the social cost of an optimal solution, and then combine them to present the following tight bound on the Price of Anarchy (PoA).

Theorem 2. *For the family $\mathcal{G}$ of resource allocation games with multiple resource classes, $PoA(\mathcal{G}) = m$.*

We show that for any number of servers, there exists a game for which the Price of Stability (PoS) is $2 - \frac{1}{m}$. This upper bound is almost matched. Our main result is a polynomial time algorithm that constructs a PNE whose social cost is at most twice the optimum. For two servers, we present a simpler algorithm and our analysis is tight.

Theorem 3. *For the family $\mathcal{G}$ of resource allocation games with multiple resource classes, $2 - \frac{1}{m} \leq PoS(\mathcal{G}) \leq m$. For two servers, $PoS(\mathcal{G}) = 3/2$.*

Our algorithms for finding a stable assignment with low social cost are based on two new methods:

1. While all the players create the same unit-load on the servers, our algorithms group the players into sets, based on their classes. An initial assignment is found by considering these sets as an instance of a multiple-knapsack packing problem with arbitrary-size elements. This method enables analysis of the assignment using known packing techniques and their properties.
2. The stabilization phase that follows the initial assignment consists of iterations in which the algorithm may reassign complete sets of players, or perform a *supervised* sequence of improving steps. The sequence is initiated by one player i, and is then limited to players of i's class who may benefit from following i by performing exactly the same migration. Analyzing the configuration after each improving step is complex; however, it is possible to analyze the effect of each supervised sequence of improving steps on the potential function and to bound the cost of an assignment derived by this method.

References

1. Anshelevich, E., Dasgupta, A., Kleinberg, J.M., Tardos, É., Wexler, T., Roughgarden, T.: The price of stability for network design with fair cost allocation. SIAM J. Comput. **38**(4), 1602–1623 (2008)
2. Feldman, M., Tamir, T.: Conflicting congestion effects in resource allocation games. J. Oper. Res. **60**(3), 529–540 (2012)
3. Vöcking, B.: In: Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V. (eds.) Algorithmic Game Theory. ch. 20: Selfish load balancing. Cambridge University Press, NY, (2007)