Dynamic Proportional Sharing: A Game-Theoretic Approach

Rupert Freeman*
Duke University
rupert@cs.duke.edu

Seyed Majid Zahedi*
Duke University
zahedi@cs.duke.edu

Vincent Conitzer
Duke University
conitzer@cs.duke.edu

Benjamin C. Lee
Duke University
benjamin.c.lee@duke.edu

ABSTRACT

Sharing computational resources amortizes cost and improves utilization and efficiency. When agents pool their resources, each becomes entitled to a portion of the shared pool. Static allocations in each round can guarantee entitlements and are strategy-proof, but efficiency suffers because allocations do not reflect variations in agents' demands for resources across rounds. Dynamic allocation mechanisms assign resources to agents across multiple rounds while guaranteeing agents their entitlements. Designing dynamic mechanisms is challenging, however, when agents are strategic and can benefit by misreporting their demands for resources.

In this paper, we show that dynamic allocation mechanisms based on max-min fail to guarantee entitlements, strategy-proofness or both. We propose the flexible lending (FL) mechanism and show that it satisfies strategy-proofness and guarantees at least half of the utility from static allocations while providing an asymptotic efficiency guarantee. Our simulations with real and synthetic data show that the performance of the flexible lending mechanism is comparable to that of state-of-the-art mechanisms, providing agents with at least 0.98x, and on average 15x, of their utility from static allocations. Finally, we propose the T-period mechanism and prove that it satisfies strategy-proofness and guarantees entitlements for $T \leq 2$.

ACM Reference Format:
Rupert Freeman, Seyed Majid Zahedi, Vincent Conitzer, and Benjamin C. Lee. 2018. Dynamic Proportional Sharing: A Game-Theoretic Approach. In *SIGMETRICS'18 Abstracts: ACM SIGMETRICS International Conference on Measurement & Modeling of Computer Systems Abstracts, June 18–22, 2018, Irvine, CA, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3219617.3219631

1 INTRODUCTION

Shared systems are defined by the competition for resources between strategic agents. In this paper, we consider a community of agents who share a non-profit system and its capital and operating costs. Sharing increases system utilization and amortizes its costs over more computation [5]. Examples include supercomputers for scientific computing [9], datacenters for Internet services [12], and clusters for academic research [4]. Note, however, that our focus excludes systems in which agents explicitly pay for time on shared computational resources (i.e., infrastructure-as-a-service).

*These authors contributed equally to this work, ordered alphabetically.

SIGMETRICS'18 Abstracts, June 18–22, 2018, Irvine, CA, USA

ACM ISBN 978-1-4503-5846-0/18/06.
https://doi.org/10.1145/3219617.3219631

Shared systems ensure fairness by allocating resources proportionally to entitlements, which specify each agent's share of system resources relative to others [8, 13]. Entitlements are dictated by exogenous factors such as agents' contributions to the shared system or priorities within the organization. A dynamic allocation mechanism should ensure agents' entitlements across time while assigning resources to computational stages that benefit most.

Guaranteeing entitlements and redistributing under-utilized resources are difficult when agents are strategic. The allocation mechanism does not know and must elicit agents' utilities, which are private information. Strategic agents act selfishly to pursue their own objectives. Agents will determine whether misreporting demands can improve their performance even at the expense of others in the system. For example, an agent is likely to over-report her demand in the current time period to obtain more resources, unless doing so leads to a reduction in the resources allocated to her in later periods.

We seek allocation mechanisms that satisfy strategy-proofness (SP), which ensures that no agent benefits by misreporting her demand for resources. Strategy-proofness is a key feature contributing to efficiency as it allows the mechanism to optimize system performance according to agents' true utilities. Without SP, agents' reports may not represent their true utility and allocating based on reported demands may not produce any meaningful performance guarantee. Moreover, strategy-proof mechanisms reduce the cognitive load on agents by eliminating the need to optimally construct resource demands or preemptively respond to misreports by other agents in the system.

Strategy-proofness is complemented by sharing incentives (SI), which ensures that agents perform at least as well as they would have by not participating in the allocation mechanism (i.e., using their own resources as a smaller, private system). With sharing incentives, agents would willingly federate their resources and manage them according to the commonly agreed upon policy. A mechanism that statically enforces entitlements in every time period satisfies strategy-proofness and sharing incentives but its efficiency is poor and fails to realize the advantages of dynamic sharing across time.

In this paper, we focus on three fundamental desiderata: sharing incentives, strategy-proofness, and efficiency. We consider agents who derive high utility per unit of resource up until some amount of resource allocation (i.e., their demand) and derive low utility

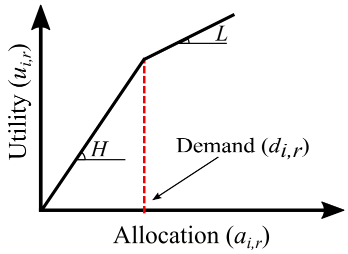

Figure 1: Users' Utility. A user derives high utility from resources up to her demand and derives low utility from resources beyond her demand.

beyond that allocation. The high-low formulation is appropriate for varied resources such as processor cores, cache and memory capacity, or virtual machines in a datacenter. For example, an agent could derive high utility when additional processors permit her to dequeue more tasks from a highly critical job. Once the job's queue is empty, she derives low utility from using additional processors to replicate tasks, which guards against stragglers or failures. In another example, an agent that is allocated more power can turn on more processors, each of which provides high utility from task parallelism. Once the agent exhausts her job's parallelism, it can use additional power to boost processor voltage and frequency for lower, non-zero utility.

We propose allocation mechanisms for dynamic proportional sharing to address limitations in existing approaches. We begin by proving that policies used in state-of-the-art schedulers [2, 3] fail to satisfy SP or SI. We then propose two alternative mechanisms. First, as our main contribution, we propose the flexible lending mechanism to satisfy SP, guarantee at least 50% of SI performance, and provide an asymptotic efficiency guarantee. The mechanism uses tokens to enable these theoretical guarantees. In practice, our simulations show that performance is comparable to that of state-of-the-art mechanisms and achieves 98% of SI performance, much better than the lower bound. Second, for situations where SI is a hard constraint, we propose the T-period mechanism to satisfy SP and SI while still outperforming static allocations.

2 BACKGROUND AND MOTIVATION

Consider a dynamic system with n agents and R discrete rounds. Agent i contributes $e_i > 0$ units of a resource at each round, which we refer to as her *endowment*. In other words, e_i is agent i's contribution to the federated system, which does not vary over time. Let $[n] = \{1, \ldots, n\}$ and $E = \sum_{i \in [n]} e_i$ denote the total number of units to be allocated at each round. At round r, agent i has a true demand of $d_{i,r} \geq 0$ units and reports a demand of $d'_{i,r} \geq 0$.

At round r, a dynamic allocation mechanism assigns each agent an allocation $a_{i,r}$ using only information from users' reported demands for the first r rounds. Agents have high (H) utility per resource up to their demand, and low (L) utility per resource that exceeds their demand. Figure 1 shows $u_{i,r}$ for user i with demand $d_{i,r}$ at round r. Agent i's overall utility after R rounds is the sum of her utility at each round.

In this paper, we focus on three main properties: sharing incentives, strategy-proofness, and efficiency. First, sharing incentives says that by participating in the mechanism, agents receive at least the utility they would have received by not participating. Next, strategy-proofness says that agents never benefit from lying about their demands. Finally, efficiency says that all resources should be allocated, and an agent with L valuation should never receive a resource while there are agents with H valuation for that resource.

In our paper, we focus on the (weighted) max-min fairness policy, which is one of the most widely used policies in computing systems. It is deployed in many state-of-the-art datacenter schedulers [2, 3], and has been extensively studied in the literature [7, 11]. A dynamic allocation mechanism could deploy the max-min policy for two different objectives: maximizing the minimum accumulated allocations up to a round, or maximizing the minimum allocation at each round, independently of previous rounds. We call the first mechanism *Dynamic Max-Min (DMM)* and the second mechanism *Static Max-Min (SMM)*. We show that SMM and DMM violate SP and SI. Indeed, in the general setting, we show that no mechanism can simultaneously satisfy efficiency and either of SP and SI.

3 PROPOSED MECHANISMS

Our aim is to design mechanisms that satisfy our game-theoretic desiderata while increasing efficiency significantly over static allocation. First, we propose the flexible lending (FL) mechanism. FL allocates exactly Re_i resources to each agent i, which is exactly her contribution to the shared pool over all R rounds. The mechanism enforces this constraint by simply removing agent i from the list of eligible agents once she has received Re_i resources in total. We keep track of the resources each agent has received with a running token count t_i, effectively 'charging' each agent a token for every resource she receives. We show that FL satisfies strategy-proofness and guarantees each agent at least 50% of her SI performance. We also show that FL provides an asymptotic efficiency guarantee.

In settings where agents require a strong guarantee in order to participate, it may be desirable to strictly enforce sharing incentives, in which case FL is not a suitable choice. To address this problem, we propose the *T-Period mechanism*. The T-Period mechanism splits the rounds into periods of length $2T$. For the first T rounds of each period, we allow the agents to 'borrow' unwanted resources from others. In the last T rounds of each period, the agents 'pay back' the resources so that their cumulative allocation across the entire period is equal to their endowment, $2Te_i$. We show that the T-Period mechanism satisfies SP and SI for $T \leq 2$. [1]

4 EVALUATION

We evaluate different mechanisms using real and synthetic benchmarks. For real benchmarks, we use a Google cluster trace [1, 10]. For synthetic benchmarks, we create random agent populations and random number of rounds.[1]

Figure 2 presents social welfare from varied allocation mechanisms for both Google and random traces normalized to the social welfare of static allocations. DMM and SMM are fully efficient mechanisms and therefore produce the same, highest social welfare.

[1]For more details on mechanisms, proofs, and experimental setup, see the full version of the paper [6].

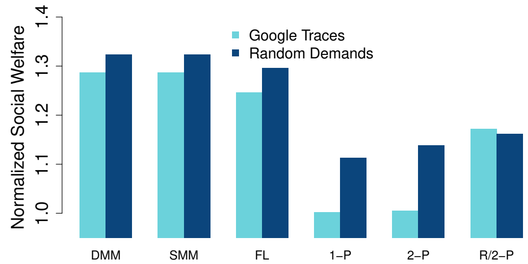

Figure 2: Normalized Social Welfare. Social welfare achieved by different dynamic allocation mechanisms normalized to that of static allocations for Google cluster traces and 100 instances of random demands.

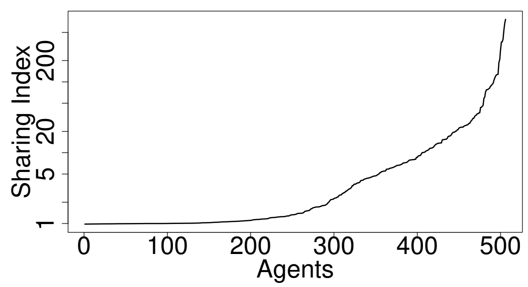

Figure 3: Sorted Sharing Index for Google Cluster Traces.

Note, however, that SMM and DMM both fail to guarantee strategy-proofness. Optimizing for reported demands may not provide high welfare according to the true demands, as reported demands and true demands may be quite different. But this is not captured in the figure, which implicitly assumes truthful reporting.

The 1-Period mechanism produces the lowest social welfare. Increasing the period length to 2 slightly improves the welfare of the T-Period mechanism. Note that both mechanisms outperform static allocations. The $R/2$-Period mechanism achieves 87% of SMM welfare for Google traces, but fails to provide strategy-proofness.

The social welfare of FL is competitive with state-of-the-art dynamic allocation mechanisms. FL achieves 97% of SMM's welfare for Google traces and 98% for random demands. In practice, strong game-theoretic desiderata do not come with high welfare costs.

To investigate violations of sharing incentives in practice, we define the *sharing index*. The sharing index of agent i is the ratio between the number of high-valued resources agent i receives under FL and under static allocations. Full sharing incentives guarantees each agent a sharing index of at least 1, while FL guarantees each agent a sharing index of at least 0.5. In practice, however, our simulations show that the sharing index is much higher.

Figure 3 shows the sharing index for all agents in the Google cluster traces, sorted in increasing order and shown on a log scale. The minimum sharing index across all agents is 0.98x, and on average agents receive 15x more utility under FL than under static allocations. In practice, the overwhelming majority of agents benefit substantially from participating in the FL mechanism.

5 CONCLUSION

We consider the problem of designing mechanisms for dynamic proportional sharing in a high-low utility model that both incentivize users to participate and share their resources (sharing incentives), as well as truthfully report their resource requirements to the system (strategy-proofness). We show that while each of these properties is incompatible with full efficiency, it is possible to satisfy both of them and still obtain some efficiency gains from sharing.

We propose the flexible lending mechanism which is strategy-proof and provides each user a theoretical guarantee of at least half her sharing incentives guarantee. While we do not guarantee full sharing incentives, we show via simulations on both real and synthetic data that in practical situations, no users are significantly worse off by participating in the sharing scheme (and the majority are vastly better off). We show that under certain assumptions, the flexible lending mechanism provides full efficiency in the large round limit, which is supported by our simulation results. We also propose the T-Period mechanism and show that it satisfies SP and SI for $T \leq 2$.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under grants CCF-1149252 (CAREER), CCF-1337215 (XPS-CLCCA), SHF-1527610, AF-1408784, CCF-133721 and IIS-1527434. This work is also supported by STARnet, a Semiconductor Research Corporation program, sponsored by MARCO and DARPA. Freeman is supported by a Facebook graduate fellowship. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of these sponsors.

REFERENCES

[1] More Google cluster data. Google research blog. (Nov. 2011). Posted at http://googleresearch.blogspot.com/2011/11/more-google-cluster-data.html.
[2] Capacity Scheduler. https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/CapacityScheduler.html. (Accessed Jan 2018).
[3] Fair Scheduler. https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/FairScheduler.html. (Accessed Jan 2018).
[4] SHARCNET. https://www.sharcnet.ca. (Accessed Jan 2018).
[5] Luiz André Barroso and Urs Hölzle. 2007. The case for energy-proportional computing. *Computer* 40, 12 (2007).
[6] Rupert Freeman, Seyed Majid Zahedi, Vincent Conitzer, and Benjamin C. Lee. 2018. Dynamic Proportional Sharing: A Game-Theoretic Approach. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2, 1 (March 2018).
[7] Ali Ghodsi, Matei Zaharia, Benjamin Hindman, Andy Konwinski, Scott Shenker, and Ion Stoica. 2011. Dominant Resource Fairness: Fair Allocation of Multiple Resource Types. In *NSDI*, Vol. 11. 24–24.
[8] J. Kay and P. Lauder. 1988. A Fair Share Scheduler. *Communications of the ACM* 31, 1 (1988), 44–55.
[9] Lawrence Berkeley National Laboratory. National Energy Research Scientific Computing Center. https://www.nersc.gov. (2017).
[10] Charles Reiss, John Wilkes, and Joseph L. Hellerstein. 2011. *Google cluster-usage traces: format + schema*. Technical Report. Google Inc., Mountain View, CA, USA. Revised 2014-11-17 for version 2.1. Posted at https://github.com/google/cluster-data.
[11] Alan Shieh, Srikanth Kandula, Albert G Greenberg, Changhoon Kim, and Bikas Saha. 2011. Sharing the Data Center Network.. In *NSDI*, Vol. 11. 23–23.
[12] Abhishek Verma, Luis Pedrosa, Madhukar Korupolu, David Oppenheimer, Eric Tune, and John Wilkes. 2015. Large-scale Cluster Management at Google with Borg. In *EuroSys*. ACM, 18:1–18:17.
[13] Carl A. Waldspurger and William E. Weihl. 1994. Lottery Scheduling: Flexible Proportional-Share Resource Management. In *OSDI*. 1–11.