PRELIMINARY PREPRINT VERSION – DO NOT CITE
This is not the version of record. The final version will appear in the Proceedings of IJCAI-ECAI 2026.

Credit Fairness: Online Fairness in Shared Resource Pools

Seyed Majid Zahedi[1] , **Rupert Freeman**[2]
[1]University of Waterloo
[2]University of Virginia
smzahedi@uwaterloo.ca, freemanr@darden.virginia.edu

Abstract

We study repeated allocation of shared resources among agents with time-varying demands and capped linear utilities. In this setting, independently maximizing the minimum utility in each round satisfies sharing incentives (agents weakly prefer participating in the mechanism to not participating), strategyproofness (agents have no incentive to misreport their demands), and Pareto efficiency. However, this max–min mechanism can lead to large disparities in the total resources received by agents, even when they have the same average demand. We introduce *credit fairness*, a property that, together with Pareto efficiency, strengthens sharing incentives by ensuring that agents who lend resources in early rounds are able to recoup them in later rounds. Credit fairness can be achieved in conjunction with either Pareto efficiency or strategyproofness individually, but we show that, under anonymity, it cannot be achieved together with both. We propose a mechanism that is credit fair and Pareto efficient, and evaluate it in a computational resource-sharing setting.

1 Introduction

In this paper, we consider a set of agents who contribute resources to a shared pool of homogeneous resources. Each agent owns a fixed endowment but faces time-varying demands. In such settings, pooling resources naturally increases utilization by accommodating agent-level demand spikes (Kelly *et al.*, 1998).

Our primary motivation is the sharing of computing resources (Ghodsi *et al.*; Stoica *et al.*, 2011; 2003), such as supercomputers for scientific computing, network bandwidth, and datacenters for internet services. More broadly, our model accommodates a wide range of applications, including shared energy systems (e.g., microgrids), transportation infrastructure (e.g., airport runways), research infrastructure (e.g., telescopes), and organizational resources shared across departments or units within a company.

In all of these settings, several requirements naturally arise as fundamental to system stability and success. *Pareto efficiency (PE)* requires that it be impossible to make some agent better off without making another agent worse off, and is essential for any system whose central goal is maximizing resource utilization. *Sharing incentives (SI)* requires that agents prefer participating in the shared system over relying solely on their own resources in every round. SI serves as both a fairness property and a stability guarantee: without SI, systems risk collapse as agents opt out. Finally, *strategyproofness (SP)* requires that agents maximize their utility by truthfully reporting their demands to the system. Because system performance can only be optimized with respect to reported demands, any discrepancy between reports and true demands can jeopardize efficiency.

While SI is a compelling property, Vuppalapati *et al.* (2023) point out that system fairness may fail even for mechanisms that satisfy SI. Focusing on resource utilization, they observe that the classic *static max-min fairness (SMMF)* mechanism—which lexicographically maximizes the minimum utility in every round—can lead to large disparities in agent utilities, despite satisfying all aforementioned properties. Intuitively, this occurs because SMMF is memoryless and therefore does not prioritize agents who contributed resources to others in earlier rounds. As a result, agents may lend resources to the system without ever being repaid. To address this issue, Vuppalapati *et al.* (2023) propose *Karma*, which guarantees agents a fraction α of their endowments in each round and allocates the remaining resources by prioritizing agents with low cumulative allocations.

Our goal in this work is to formally capture, via an axiomatic property, the sense in which SMMF is unfair. To this end, we introduce *credit fairness*. Intuitively, credit fairness posits the existence of an accounting framework (a "credit system") that tracks lending and borrowing and ensures that agents who are net donors are not deprioritized in favor of agents who are net borrowers. Moreover, when combined with PE, credit fairness constitutes a strengthening of SI: it preserves the stability interpretation of SI while enhancing it from a fairness perspective. This strengthening comes at a cost: under anonymity, credit fairness is incompatible with simultaneously satisfying PE and SP.

To our knowledge, all existing mechanisms in this setting violate credit fairness, including Karma. We design a simple mechanism, LENDRECOUP, that is both credit fair and PE. Additionally, we show that LENDRECOUP is *online strategyproof (OSP)* (Aleksandrov and Walsh, 2019b), which as-

Mechanism	PE	SP	OSP	SI	Credit Fair
SMMF	Yes	Yes	Yes	Yes	No
DMMF	Yes	No	Yes	No	No
Karma ($\alpha \in (0, 1)$)	Yes	No	?	No	No
Karma ($\alpha = 1$)	Yes	No	?	Yes	No
LENDRECOUP	Yes	No	Yes	Yes	Yes

Table 1: **Summary of properties** satisfied by static max-min fairness, dynamic max-min fairness, Karma, and LENDRECOUP.

sumes that agents do not account for future rounds when strategizing. While this is a relatively weak guarantee, it is, in some sense, the strongest one can hope for, as our impossibility result rules out full SP. Table 1 summarizes the properties satisfied by existing mechanisms.[1]

Finally, using data from a real-world environment, we show that LENDRECOUP's performance is comparable to or better than that of other mechanisms across a range of fairness and efficiency metrics. Combined with the formal guarantee of credit fairness, these results suggest that LENDRECOUP is a compelling candidate for practical deployment.

2 Related Work

In recent years, there has been renewed interest in repeated allocation mechanisms in dynamic settings where agents have time-varying demands (Freeman *et al.*; Hossain; Li *et al.*; Vardi *et al.*; Vuppalapati *et al.*; Fikioris *et al.*; Fikioris *et al.*; Cookson *et al.*, 2018; 2019; 2021; 2022; 2023; 2024; 2025; 2025). The work most closely related to ours is that of Freeman *et al.* (2018), who study the online allocation of multiple resource units from a shared pool among agents who both contribute to and draw from the pool over multiple rounds. Agents are assumed to have piecewise-linear utilities that are high up to a demand and low beyond it. They consider two settings: (I) zero utility beyond demands and (II) strictly positive utility beyond demands. For (I), they analyze two max–min fair mechanisms: dynamic max–min fair (DMMF), which lexicographically maximizes the minimum cumulative utility up to the current round, and static max–min fair (SMMF), which lexicographically maximizes the minimum utility in the current round. They show that SMMF satisfies SP, SI, and PE, while DMMF violates SP and SI. For (II), they show that no online mechanism can satisfy PE along with either SI or SP.

Another closely related work is that of Vuppalapati *et al.* (2023), who study setting (I) of Freeman *et al.* (2018) and propose *Karma*, a mechanism parameterized by $0 \leq \alpha \leq 1$, which specifies the fraction of each agent's endowment that is guaranteed when demanded. The remaining resources, together with donations from agents whose demand falls below their guaranteed share, are allocated to agents with excess demand by prioritizing those with the smallest cumulative allocations. Agents gain credits when others borrow their donated resources and lose credits when borrowing from the pool. They derive theoretical guarantees only for $\alpha = 0$, in which case *Karma* reduces to DMMF, violating SP and SI.

Repeated allocation has also been studied in settings where indivisible items arrive over time and must be allocated immediately without revocation. Zeng and Psomas (2020) and Benadè *et al.* (2024) study the trade-off between fairness and efficiency in this setting, while Aleksandrov and Walsh (2019a) examine online fair division under monotonicity constraints. Aleksandrov and Walsh (2019b) show that, for agents with additive utilities, SP, envy-freeness, and PE cannot be simultaneously achieved against worst-case adversaries. Benadè *et al.* (2018) provide a deterministic algorithm with vanishing envy over time. Benadè *et al.* (2022) study a partial information setting, where ordinal but not cardinal information is elicited and values are drawn from i.i.d. distributions. Finally, Igarashi *et al.* (2024) study the repeated allocation of indivisible goods and chores, providing guarantees on both per-round and cumulative fairness and efficiency.

3 Preliminaries

For $k \in \mathbb{N}$, we denote $[k] \coloneqq \{1, \ldots, k\}$. Consider a system with n agents and discrete rounds $t = 1, 2, \ldots$. (All definitions apply to any finite prefix of rounds.) At each round, each agent i contributes a fixed *endowment* of $e_i > 0$ units, and the total endowment is $E = \sum_i e_i$. We denote by $E_{-i} = \sum_{j \neq i} e_j$ the total number of units not including i's endowment. We assume a dynamic setting in which agents have time-varying demands. At round t, agent i has a true demand of $d_{i,t} \geq 0$ units and reports a demand of $d'_{i,t} \geq 0$. Let $\boldsymbol{d}_{i,1:t} = (d_{i,t'})_{1 \leq t' \leq t}$ and $\boldsymbol{d}'_{i,1:t} = (d'_{i,t'})_{1 \leq t' \leq t}$ denote the vectors of agent i's actual and reported demands up to round t, respectively. Further, let $\boldsymbol{d}'_{-i,1:t}$ denote the reported demands of all agents other than i up to round t. At every round t, an *online allocation mechanism* M (henceforth simply *mechanism*) assigns each agent an allocation $a^M_{i,t}(\boldsymbol{d}'_{1,1:t}, \ldots, \boldsymbol{d}'_{n,1:t}) = a^M_{i,t}(\boldsymbol{d}'_{i,1:t}, \boldsymbol{d}'_{-i,1:t})$ with $\sum_i a^M_{i,t}(\boldsymbol{d}'_{i,1:t}, \boldsymbol{d}'_{-i,1:t}) \leq E$. We will typically write simply $a_{i,t}$ when the exact mechanism and the demands are clear from context. Let $\boldsymbol{a}^M_{i,1:t}(\boldsymbol{d}'_{i,1:t}, \boldsymbol{d}'_{-i,1:t})$ (often simply $\boldsymbol{a}_{i,1:t}$) denote the vector of agent i's allocations up to round t. Agents derive one unit of utility for every unit of resource they receive up to their demand and do not value additional units; that is, the utility of agent i with demand $d_{i,t}$ at round t is $u_{i,t}(a_{i,t}, d_{i,t}) = \min(d_{i,t}, a_{i,t})$. We additionally assume that utilities are additive over rounds, so that agent i's total utility over the first t rounds for an allocation $\boldsymbol{a}_{i,1:t}$ is $U_{i,t}(\boldsymbol{a}_{i,1:t}, \boldsymbol{d}_{i,1:t}) = \sum_{t'=1}^{t} u_{i,t'}(a_{i,t'}, d_{i,t'})$. For notational simplicity, we write $u_{i,t}(a_{i,t})$ or $U_{i,t}(\boldsymbol{a}_{i,1:t})$ when the dependence on $d_{i,t}$ or $\boldsymbol{d}_{i,1:t}$ is clear. It will often be convenient to reason about an agent's utility with respect to their reported, rather than true, demand; for this purpose, we use the shorthand $u'_{i,t}(a_{i,t}) = u_{i,t}(a_{i,t}, d'_{i,t})$ and $U'_{i,t}(\boldsymbol{a}_{i,1:t}) = U_{i,t}(\boldsymbol{a}_{i,1:t}, \boldsymbol{d}'_{i,1:t})$. Although this quantity does not represent an agent's actual utility, it appears often because our definition of credit fairness will disregard incentive

[1] Note that DMMF corresponds to Karma with parameter $\alpha = 0$. For Karma with $0 < \alpha < 1$, we report "Yes" when the property holds for all values of α and "No" when there exists some α for which the property is violated. Since no formal theoretical analysis exists for Karma with $\alpha > 0$, we report only properties that are straightforward to verify without proof; whether Karma satisfies OSP remains an open question.

considerations and implicitly assume that reported demands equal true demands, as is standard for fairness properties.

Following prior work (Freeman *et al.*, 2018), we allow the allocations $a_{i,t}$ to be real-valued even though resource units and demands are discrete. We interpret real-valued allocations as being implemented via randomization where agent i is allocated $a_{i,t}$ units in expectation.

Properties of Mechanisms. In addition to the new property that we will define later, we focus on three fundamental desiderata: Pareto efficiency (PE), sharing incentives (SI), and strategyproofness (SP). PE requires that if any agent receives fewer units than their demand, then all units must be allocated to agents who derive positive utility from them, i.e., the sum of agents' utilities equals E.

Definition 1. A mechanism M is *Pareto efficient* if, for every round t and every report profile $(\boldsymbol{d}'_{1,1:t}, \dots, \boldsymbol{d}'_{n,1:t})$, the following holds: whenever there exists an agent i such that $a^M_{i,t}(\boldsymbol{d}'_{i,1:t}, \boldsymbol{d}'_{-i,1:t}) < d'_{i,t}$, then it must be the case that:

$$\sum_j \min\bigl(a^M_{j,t}(\boldsymbol{d}'_{j,1:t}, \boldsymbol{d}'_{-j,1:t}),\ d'_{j,t}\bigr) = E.$$

SI requires that agents receive at least as much utility from (truthfully) participating in the system as they would have received by not participating.

Definition 2. A mechanism M satisfies *sharing incentives* if, for every round t, every agent i, all true demands $\boldsymbol{d}_{i,1:t}$, and any report profile $\boldsymbol{d}'_{-i,1:t}$ of the other agents, it holds that:

$$U_{i,t}\bigl(\boldsymbol{a}^M_{i,1:t}(\boldsymbol{d}_{i,1:t}, \boldsymbol{d}'_{-i,1:t})\bigr) \ \geq\ U_{i,t}(\boldsymbol{e}_i),$$

where $\boldsymbol{e}_i = (e_i, \dots, e_i)$ denotes the allocation in which agent i receives their endowment in every round.

SP requires that agents never benefit from misreporting.

Definition 3. A mechanism M is *strategy-proof* if, for every agent i, every round t, all true demands $\boldsymbol{d}_{i,1:t}$, all report profiles $\boldsymbol{d}'_{i,1:t}$ of agent i, and any report profile $\boldsymbol{d}'_{-i,1:t}$ of the other agents, it holds that:

$$U_{i,t}\bigl(\boldsymbol{a}^M_{i,1:t}(\boldsymbol{d}_{i,1:t}, \boldsymbol{d}'_{-i,1:t})\bigr) \ \geq\ U_{i,t}\bigl(\boldsymbol{a}^M_{i,1:t}(\boldsymbol{d}'_{i,1:t}, \boldsymbol{d}'_{-i,1:t})\bigr).$$

Finally, *online strategyproofness (OSP)* (Aleksandrov and Walsh, 2019b) requires that, fixing reports of all agents in rounds $1, \dots, t-1$, no agent can increase their utility in round t by misreporting.

Definition 4. A mechanism M is *online strategy-proof* if, for every agent i, every round t, all demands $\boldsymbol{d}_{i,1:t}$, all report profiles $\boldsymbol{d}'_{i,1:t}$ of agent i such that $\boldsymbol{d}_{i,1:t-1} = \boldsymbol{d}'_{i,1:t-1}$, and any report profile $\boldsymbol{d}'_{-i,1:t}$ of the other agents, it holds that:

$$u_{i,t}\left(a^M_{i,t}(\boldsymbol{d}_{i,1:t}, \boldsymbol{d}'_{-i,1:t})\right) \ \geq\ u_{i,t}\left(a^M_{i,t}(\boldsymbol{d}'_{i,1:t}, \boldsymbol{d}'_{-i,1:t})\right).$$

4 Credit Fairness

It is known that SMMF satisfies PE, SI, and SP (Freeman *et al.*, 2018). However, as Vuppalapati *et al.* (2023) note, SMMF can produce highly inequitable allocations in terms of the number of resources received—as large as a factor of $\Omega(n)$—even between users with the same average demand. We illustrate this shortcoming with the following example, which is heavily inspired by Vuppalapati *et al.*'s proof of the aforementioned result.

Example 1. Suppose there are 3 rounds and 3 agents, each with endowment $e_i = 1$, and the following demands:

	$t=1$	$t=2$	$t=3$
Agent 1	2	2	2
Agent 2	2	2	2
Agent 3	0	0	6.

Note that the total demand of all three agents is the same. Under SMMF, the resulting allocations are as follows:

	$t=1$	$t=2$	$t=3$
Agent 1	1.5	1.5	1
Agent 2	1.5	1.5	1
Agent 3	0	0	1.

Agents 1 and 2 each receive a utility of 4, while agent 3 receives a utility of 1.

Agent 3 could argue that, having lent their endowment to other agents in the first two rounds, they should have the right to reclaim those resources in round 3 when they need them. However, because SMMF is memoryless, it denies this possibility. In this section, we formally capture the undesirable behavior of SMMF in this example by introducing *credit fairness*, an axiomatic property that reflects the idea that agents should have priority over resources they previously lent.

4.1 Formal Definition

To define credit fairness, we require the notion of a *credit system*. A credit system $(c_{i,t})_{i\in[n],t\in\mathbb{N}}$ is simply a balance of credits for every agent. We denote by $c_{i,t} \in \mathbb{R}$ the number of credits that agent i has at the start of round t, and require that $c_{i,1} = 0$ for every agent i. Let $\Delta c_{i,t} \triangleq c_{i,t+1} - c_{i,t}$. Furthermore, for every agent i and round t, let $A_{-i,t} = A^M_{-i,t}(\boldsymbol{d}'_{1,1:t}, \dots, \boldsymbol{d}'_{n,1:t}) \triangleq \sum_{j\neq i} a_{i,t}$ denote the total round t allocation of agents other than i under mechanism M.

Definition 5. A mechanism M is *credit fair* if there exists a credit system $(c_{i,t})_{i\in[n],t\in\mathbb{N}}$ for which the following properties hold for all agents i and rounds t:

(CF1) $\min(0, e_i - u'_{i,t}(a_{i,t})) \ \leq\ \Delta c_{i,t} \ \leq\ \max(0, e_i - u'_{i,t}(a_{i,t}))$;

(CF2) If $\sum_{j\neq i} u'_{j,t}(a_{j,t}) > E_{-i}$, then $\Delta c_{i,t} \geq \sum_{j\neq i} u'_{j,t}(a_{j,t}) - E_{-i}$;

(CF3) $\sum_{i\in[n]} \Delta c_{i,t} \leq 0$;

(CF4) $a_{i,t} \geq \min(d'_{i,t}, e_i + \min(0, c_{i,t}))$; and

(CF5) If $a_{j,t} > \max(0, e_j + c_{j,t})$ for some agent j, then for all agents i it holds that either $a_{i,t} \geq d'_{i,t}$ or $a_{i,t} \geq e_i + c_{i,t}$.

Let us unpack this definition. Condition (CF1) states that agents who receive less than e_i utility (informally, agents donating resources to the system) should not lose credits but may gain up to $e_i - u'_{i,t}(a_{i,t})$. Likewise, agents who receive more than e_i utility (informally, borrowers) should not gain credits but may lose up to $u'_{i,t}(a_{i,t}) - e_i$. Agents whose utility equals their endowment neither gain nor lose credits. Although this condition does not require a one-to-one correspondence between credits and resources, it bounds how far credits can diverge from allocations. At a high level, it justifies interpreting positive credits as claims on resources and negative credits as obligations.

Condition (CF2) requires that if an agent i lends resources to others, i.e., the other agents collectively receive utility exceeding the resources they contribute, then agent i must gain at least one unit of credit for each such resource. This captures the idea that agents should be rewarded when they are pivotal to others receiving valued resources.

Condition (CF3) requires that the credit system be deflationary: the total number of credits cannot increase over time. Since credits represent claims on resources, allowing their total to grow would imply that agents could collectively obtain resources without anyone bearing a corresponding cost.

To understand condition (CF4), consider an agent with $d_{i,t} > e_i$. If the agent has positive credits, they must receive at least their endowment e_i. If the agent has negative credits, they are guaranteed only $e_i + c_{i,t}$. Intuitively, the system may reclaim from the agent up to the number of resources equal to the credit they owe.

Finally, condition (CF5) relies on the notion of a *credit-adjusted endowment*, defined as $e_i + c_{i,t}$, which reflects both the agent's baseline endowment and whether they are a net borrower or lender. The condition requires that if some agent j receives strictly more than their credit-adjusted endowment in a round, then every other agent with sufficient demand must receive at least their own credit-adjusted endowment. Interpreting the credit-adjusted endowment as an agent's "fair share," this ensures that no agent receives more than their fair share while another receives strictly less.

In the full version of the paper (Zahedi and Freeman, 2026), we show that none of CF1–CF5 is implied by the remaining four conditions.

In summary, credit fairness establishes a memory-based system in which credits act as a ledger of past deviations from an agent's endowment: agents who receive less accumulate limited future claims, while those who receive more incur limited future obligations. Rather than guaranteeing a fixed minimum share, as adopted by mechanisms such as Karma, credit fairness enforces a dynamic notion of equity in which an agent's entitlement adapts to their credit balance. Agents with positive credits are guaranteed at least their endowment, while agents in debt have their guaranteed minimum reduced according to what they owe.

4.2 Properties

We begin with a useful lemma that characterizes credit updates for any credit-fair and Pareto-efficient mechanism in settings where resources are overdemanded. The proof of this lemma and all other omitted proofs appear in the full version of the paper (Zahedi and Freeman, 2026).

Lemma 1. *Fix a round t with $\sum_j d'_{j,t} \geq E$. Let M be a mechanism that is* credit fair *and* Pareto efficient*. Then, for every agent $i \in [n]$, it holds that $\Delta c_{i,t} = e_i - a_{i,t}$.*

We now show that, when combined with Pareto efficiency, credit fairness implies sharing incentives. The proof is omitted for space constraints, but the main idea is to show that if an agent receives less than their SI baseline utility $u'_{i,t}(e_i)$ in some round, then that agent must have nonpositive credits, even after the round-t allocations, i.e., $c_{i,t+1} \leq 0$. This fact is sufficient to guarantee that the agent's cumulative utility still does not fall below their SI baseline.

Theorem 1. *Let M be a credit-fair and Pareto efficient mechanism. Then M satisfies sharing incentives.*

As a corollary of Theorem 1, any Pareto efficient mechanism that violates sharing incentives cannot be credit fair. Since DMMF is Pareto efficient but violates sharing incentives, it follows that DMMF is not credit fair.

Corollary 1. *The dynamic max-min fair mechanism violates credit fairness.*

Furthermore, we show that SMMF, despite satisfying sharing incentives and Pareto efficiency, violates credit fairness, which further validates the necessity of our definition.

Proposition 1. *The static max-min fair mechanism violates credit fairness.*

To our knowledge, SMMF is the only known mechanism that satisfies SP, PE, and SI, yet it violates credit fairness. This naturally raises the question of whether there exists a mechanism that satisfies the stronger combination of SP, PE, and credit fairness. Our next result shows that the answer is no, at least under the mild restriction of anonymity. A mechanism is *anonymous* if permuting the agents results in the allocation being permuted in the same way; in other words, the mechanism does not depend on the identities of the agents. Whether anonymity can be removed from the following theorem is an open question.

Theorem 2. *No mechanism is anonymous, strategyproof, credit fair, and Pareto efficient.*

Proof. Consider $n = 3$ agents, each with endowment of $e_i = 1$. We examine the first 5 rounds with the following demands:

	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$
Agent 1	1	2	0	0	3
Agent 2	3	0	1	1	2
Agent 3	0	2	2	2	0.

Let M be credit fair, PE, and anonymous (and therefore also satisfies SI). Then M must produce the following allocations:

	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$
Agent 1	1	1	0	0	2
Agent 2	2	0	1	1	1
Agent 3	0	2	2	2	0,

and the following credit balances:

	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$
Agent 1	0	0	0	1	2
Agent 2	0	-1	0	0	0
Agent 3	0	1	0	-1	-2.

Note that, given the demands and allocations, these credit balances are implied by Lemma 1.

The first-round allocations are implied by SI and PE. In the second round, SI requires that both agents 1 and 3 receive at least one unit, and (CF5) then requires that the additional unit be allocated to agent 3. The third- and fourth-round allocations are dictated by PE. Finally, the fifth-round allocation is implied by (CF5), which does not allow agent 2 to receive more than one unit unless agent 1 receives at least three.

Now suppose agent 1 instead reports a demand of 0 in the first round. In this case, the allocations must be:

	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$
Agent 1	0	1.5	0	0	3
Agent 2	3	0	1	1	0
Agent 3	0	1.5	2	2	0,

with the following credits:

	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$
Agent 1	0	1	0.5	1.5	2.5
Agent 2	0	-2	-1	-1	-1
Agent 3	0	1	0.5	-0.5	-1.5.

To verify the allocations, observe that the allocations in the first, third, and fourth rounds are required by PE, the second-round allocation follows from anonymity, and the fifth-round allocation is implied by (CF5). As a result of the misreport, agent 1's total utility increases by 0.5, from 4 to 4.5, demonstrating a profitable deviation. Therefore, M is not SP. ☐

While SP, PE, and credit fairness cannot all be satisfied simultaneously by an anonymous mechanism, any two of them can. SMMF is SP and PE but not credit fair. The static mechanism that allocates each agent their endowment $a_{i,t} = e_i$ at each round is SP and credit fair (SP is obvious; see [Zahedi and Freeman, 2026] for a proof of credit fairness). In the next section we will introduce a mechanism that is credit fair and PE.

5 LendRecoup Mechanism

We present a mechanism, LendRecoup, that is credit fair and PE. At each round, LendRecoup allocates resources by first satisfying demands as much as possible up to agents' credit-adjusted endowment $\max(0, e_i + c_{i,t})$. If resources remain after every agent receives $\min(d'_{i,t}, \max(0, e_i + c_{i,t}))$, the surplus is distributed among agents with remaining demand, prioritizing agents with low cumulative allocations. If resources still remain after all agents receive their demands, then remaining resources are allocated to agents with low demand in the current round. Credits are updated in a one-to-one correspondence with resource transfers: an agent gains (loses) one credit for each unit of resource lent to (borrowed from) the system.

To formally specify LendRecoup, we use the Proportional Sharing With Constraints (PSWC) procedure of Freeman *et al.* (2018). This procedure takes as input a total resource amount A, agent weights $\boldsymbol{w}$, and agent-specific lower and upper bounds $\boldsymbol{m}$ (for "minimum") and $\boldsymbol{l}$ (for "limit"). It outputs per-agent allocations by finding a scalar x such that each agent i receives xw_i units whenever $m_i \le xw_i \le l_i$; otherwise, agent i receives m_i if $xw_i < m_i$, or l_i if $xw_i > l_i$. The PSWC procedure can be implemented in $O(n \log n)$ time using the Divvy algorithm (Gulati *et al.*, 2012).

Pseudocode for LendRecoup is given in Algorithm 1. To implement prioritization of agents with low cumulative allocations, we conceptually "reallocate" all resources from previous rounds using the PSWC procedure, with minimum allocations set to the allocations already made. This allows any remaining uncommitted resources at round t to be used to equalize cumulative allocations as much as possible.

Algorithm 1 LendRecoup

Require: Set of agents $[n]$, endowments $\boldsymbol{e} = (e_i)_i$ (with $E = \sum_i e_i$), and online reports $\boldsymbol{d}'_t = (d'_{i,t})_i$ revealed in each round.
Ensure: For each round t, allocations $\boldsymbol{a}_t = (a_{i,t})_i$ and credits $\boldsymbol{c}_{t+1} = (c_{i,t+1})_i$.

1: ▷ *Initializing credits to zero* ◁
2: $\boldsymbol{c}_1 \leftarrow \boldsymbol{0}$
3: **for** $t = 1$ **to** T **do**
4: **if** $\sum_i d'_{i,t} \le E$ **then** ▷ *No shortage*
5: ▷ *Call PSWC with minima at demands* ◁
6: $\boldsymbol{a}_t \leftarrow \texttt{PSWC}(A = E, \boldsymbol{w} = \boldsymbol{e}, \boldsymbol{m} = \boldsymbol{d}'_t, \boldsymbol{l} = +\boldsymbol{\infty})$
7: **else** ▷ *Shortage*
8: ▷ *Cap demands at credit-adjusted endowments* ◁
9: $\bar{\boldsymbol{d}}_t \leftarrow \min(\boldsymbol{d}'_t, \max(\boldsymbol{0}, \boldsymbol{e} + \boldsymbol{c}_t))$
10: **if** $E \le \sum_i \bar{d}_{i,t}$ **then**
11: ▷ *Call PSWC up to capped demands* ◁
12: $\boldsymbol{a}_t \leftarrow \texttt{PSWC}\left(A = E, \boldsymbol{w} = \boldsymbol{e}, \boldsymbol{m} = \boldsymbol{0}, \boldsymbol{l} = \bar{\boldsymbol{d}}_t\right)$
13: **else**
14: ▷ *Prioritize low cumulative allocations* ◁
15: $\boldsymbol{m} \leftarrow \sum_{t'=1}^{t-1} \boldsymbol{a}_{t'} + \bar{\boldsymbol{d}}_t$
16: $\boldsymbol{l} \leftarrow \sum_{t'=1}^{t-1} \boldsymbol{a}_{t'} + \boldsymbol{d}'_t$
17: $\tilde{\boldsymbol{a}}_t \leftarrow \texttt{PSWC}(A = tE, \boldsymbol{w} = \boldsymbol{e}, \boldsymbol{m}, \boldsymbol{l})$
18: $\boldsymbol{a}_t \leftarrow \tilde{\boldsymbol{a}}_t - \sum_{t'=1}^{t-1} \boldsymbol{a}_{t'}$
19: ▷ *Update credits based on lending and borrowing* ◁
20: $\boldsymbol{c}_{t+1} \leftarrow \boldsymbol{c}_t + \boldsymbol{e} - \boldsymbol{a}_t$

Example 2. Consider the instance from Example 1. On this example, LendRecoup makes the following allocations:

	$t=1$	$t=2$	$t=3$
Agent 1	1.5	1.5	0
Agent 2	1.5	1.5	0
Agent 3	0	0	3,

with the following credit balances:

	$t=1$	$t=2$	$t=3$
Agent 1	0	-0.5	-1
Agent 2	0	-0.5	-1
Agent 3	0	1	2.

In all rounds the algorithm is in the shortage branch (Line 7). The critical round is $t = 3$, in which we have credit-adjusted endowments $\bar{d}_{1,3} = \bar{d}_{2,3} = 0$ and $\bar{d}_{3,3} = 3$, yielding $\sum_i \bar{d}_{i,t} = 3 \ge E = 3$. Therefore, LendRecoup runs PSWC with minimum allocations 0 and maximum allocations $\bar{\boldsymbol{d}}_t$, resulting in all 3 resources being allocated to agent 3.

LendRecoup is PE, since it never allocates resources that are not valued unless all demands are exhausted.

Proposition 2. LendRecoup *is Pareto efficient.*

We next show that LendRecoup is credit fair.

Theorem 3. LendRecoup *satisfies credit fairness.*

Proof. We examine the five conditions individually.

Condition (CF1) holds in the shortage case by the definition of the credit update and the fact that $a_{i,t} = u'_{i,t}(a_{i,t})$ in this case; hence, $\Delta c_{i,t} = e_i - a_{i,t}$ for all i. In the no-shortage case, the same argument applies to agents with $a_{i,t} = d'_{i,t}$. Otherwise, note that $a_{i,t} > d'_{i,t}$ implies $a_{i,t} \leq e_i$, since the PSWC call never reaches $x > 1$ (as $x > 1$ would allocate more than e_i resources to every agent, violating the capacity constraint). For such agents, we have $0 \leq \Delta c_{i,t} = e_i - a_{i,t} \leq e_i - u'_{i,t}(a_{i,t})$.

Condition (CF2). Fix any agent i and round t. Observe that: $\sum_{j \neq i} u'_{j,t}(a_{j,t}) \leq \sum_{j \neq i} a_{j,t} = \sum_j a_{j,t} - a_{i,t} = E - a_{i,t}$. Rearranging and substituting $\Delta c_{i,t} = e_i - a_{i,t}$ gives: $\sum_{j \neq i} u'_{j,t}(a_{j,t}) - E_{-i} \leq e_i - a_{i,t} = \Delta c_{i,t}$.

Condition (CF3) holds trivially since $\sum_j \Delta c_{j,t} = \sum_j (e_j - a_{j,t}) = E - \sum_j a_{j,t} = 0$.

Condition (CF4). Fix an agent i and round t. We consider the following two cases.

Case 1: $\sum_i d'_{i,t} \leq E$. In this case, Algorithm 1 calls PSWC with minima $\boldsymbol{m} = \boldsymbol{d'_t}$, which implies $a_{i,t} \geq d'_{i,t} \geq \min\big(d'_{i,t},\, e_i + \min(0, c_{i,t})\big)$, as required by (CF4).

Case 2: $\sum_i d'_{i,t} > E$. Define $\kappa_{i,t} := \max(0, e_i + c_{i,t})$ so that $\bar{d}_{i,t} := \min(d'_{i,t}, \kappa_{i,t})$. We consider two subcases.

Case 2a: $E \leq \sum_j \bar{d}_{j,t}$. The mechanism calls PSWC with weights $\boldsymbol{w} = \boldsymbol{e}$, minima $\boldsymbol{m} = \boldsymbol{0}$, and upper limits $\boldsymbol{l} = \boldsymbol{\bar{d}_t}$. Without upper limits, PSWC would return the proportional allocation $a_{i,t} = e_i$ (since $A = E = \sum_i e_i$). With the upper limits, PSWC can only reduce the allocation of agents whose proportional share exceeds their cap, and redistribute any leftover to agents whose cap exceeds their proportional share. In particular, we have $a_{i,t} \geq \min(e_i, \bar{d}_{i,t})$ for all agents i. Now consider $c_{i,t} \geq 0$ and $c_{i,t} < 0$ separately. If $c_{i,t} \geq 0$, then $\kappa_{i,t} \geq e_i$, hence $\bar{d}_{i,t} = \min(d'_{i,t}, \kappa_{i,t}) \geq \min(d'_{i,t}, e_i)$. Therefore, we have:

$$a_{i,t} \geq \min(e_i, \bar{d}_{i,t}) \geq \min(e_i, d'_{i,t}) = \min\big(d'_{i,t},\, e_i + \min(0, c_{i,t})\big).$$

If $c_{i,t} < 0$ and $e_i + c_{i,t} < 0$, the desired inequality holds trivially since allocations are nonnegative. If $c_{i,t} < 0$ and $e_i + c_{i,t} \geq 0$, then $\kappa_{i,t} = e_i + c_{i,t}$, and therefore $\bar{d}_{i,t} = \min(d'_{i,t}, e_i + c_{i,t})$. This implies $\bar{d}_{i,t} \leq e_i$, and hence $\min(e_i, \bar{d}_{i,t}) = \bar{d}_{i,t}$. As a result, we have:

$$a_{i,t} \geq \bar{d}_{i,t} = \min(d'_{i,t},\, e_i + c_{i,t}) = \min\big(d'_{i,t},\, e_i + \min(0, c_{i,t})\big).$$

Case 2b: $E > \sum_j \bar{d}_{j,t}$. In this case, the mechanism calls PSWC with minima set to $\boldsymbol{m} = \sum_{t'=1}^{t-1} \boldsymbol{a_{t'}} + \boldsymbol{\bar{d}_t}$, which guarantees that the cumulative allocation after round t is at least $\sum_{t'=1}^{t-1} a_{i,t'} + \bar{d}_{i,t}$. Thus $a_{i,t} \geq \bar{d}_{i,t} = \min(d'_{i,t}, \kappa_{i,t})$. If $c_{i,t} \geq 0$, then $\kappa_{i,t} \geq e_i$, which implies $\bar{d}_{i,t} \geq \min(d'_{i,t}, e_i) = \min(d'_{i,t}, e_i + \min(0, c_{i,t}))$. If $c_{i,t} < 0$ and $e_i + c_{i,t} < 0$, the bound is trivial. If $c_{i,t} < 0$ and $e_i + c_{i,t} \geq 0$, then $\kappa_{i,t} = \max(0, e_i + c_{i,t})$, and the same argument as above yields $a_{i,t} \geq \min(d'_{i,t}, e_i + c_{i,t}) = \min(d'_{i,t}, e_i + \min(0, c_{i,t}))$.

Condition (CF5). Fix a round t and suppose there exists an agent j such that: $a_{j,t} > \max(0, e_j + c_{j,t}) = \kappa_{j,t}$. We must show that for every agent i, either $a_{i,t} \geq d'_{i,t}$ or $a_{i,t} \geq e_i + c_{i,t}$.

Case 1: $\sum_i d'_{i,t} \leq E$. In this case, the mechanism allocates $a_{i,t} \geq d'_{i,t}$ to all agents, so (CF5) holds trivially.

Case 2: $\sum_i d'_{i,t} > E$ *and* $E \leq \sum_i \bar{d}_{i,t}$. Here, PSWC is called with upper limits $\boldsymbol{l} = \boldsymbol{\bar{d}_t}$. Therefore, for every agent i we have $a_{i,t} \leq \bar{d}_{i,t} \leq \kappa_{i,t}$. Therefore, the premise $a_{j,t} > \kappa_{j,t}$ can never hold, and (CF5) holds vacuously.

Case 3: $\sum_i d'_{i,t} > E$ *and* $E > \sum_i \bar{d}_{i,t}$. In this case, as shown above, the minimum constraints guarantee that every agent k receives at least $\bar{d}_{k,t} = \min(d'_{k,t}, \kappa_{k,t})$ units. Now fix any agent i. If $d'_{i,t} \leq \kappa_{i,t}$, then $\bar{d}_{i,t} = d'_{i,t}$, and hence $a_{i,t} \geq d'_{i,t}$. Otherwise, $d'_{i,t} > \kappa_{i,t}$, in which case $\bar{d}_{i,t} = \kappa_{i,t}$ and $a_{i,t} \geq \bar{d}_{i,t} = \kappa_{i,t} = \max(0, e_i + c_{i,t}) \geq e_i + c_{i,t}$. Therefore, for every i, either $a_{i,t} \geq d'_{i,t}$ or $a_{i,t} \geq e_i + c_{i,t}$. $\square$

Since LENDRECOUP is anonymous, Theorem 2 implies that it cannot be strategyproof. However, we can show that it satisfies the weaker notion of *online strategyproofness*.

Theorem 4. LENDRECOUP *is online strategy-proof.*

6 Experiments

We compare three alternatives against LENDRECOUP: (i) static max-min fairness (SMMF), (ii) dynamic max-min fairness (DMMF), and (iii) Karma (with $\alpha = 0.5$). For Karma, we choose $\alpha = 0.5$ because Vuppalapati *et al.* (2023) use that value of α in their own empirical evaluation. Recall that Karma with $\alpha = 0$ corresponds to DMMF.

6.1 Workloads

We evaluate mechanisms over 50 independent simulation repetitions. In each repetition, we simulate a system with 50 agents over 500 rounds. To generate demand profiles, we use Google cluster traces (Wilkes; Reiss *et al.*, 2011; 2011), similar to [Freeman *et al.*, 2018]. These traces record task-processing events on a 12.5k-machine cluster over roughly one month in May 2011. Each task submission specifies resource demands requested by the submitting agent, including CPU, memory, and disk space. Demands are normalized relative to the maximum capacity of each resource across all machines in the traces. We focus exclusively on CPU requests.

Following [Freeman *et al.*, 2018], we divide time into 15-minute rounds and define each agent's demand in a round as the total CPU requested by tasks submitted during that interval. After processing the traces, we remove agents with constant demands or with average demand below a small threshold. Each remaining agent is assigned an endowment equal to its average demand across all rounds. This reflects real-world settings where agents contribute resources roughly proportional to their long-term demand and rely on sharing during demand spikes.

6.2 Evaluation Metrics

We evaluate mechanisms using agents' cumulative utilities U_i. Let U_i^{static} denote the utility agent i receives under the *static* allocation, where each agent obtains their endowment in every round. We consider the following metrics.

Mechanism	**NW** (std)	**Min SIx** (std)	**% SI Vio.** (std)	**WMM** (std)	**NMM** (std)	**WEq** (std)	**NEq** (std)
LENDRECOUP	1.018 (0.54)	**1.00** (0.00)	**0.0** (0)	0.26 (0.15)	0.013 (0.01)	0.26 (0.15)	**0.87** (0.09)
DMMF	**1.019** (0.54)	0.96 (0.04)	35.8 (12.5)	**0.40** (0.17)	0.010 (0.01)	**0.42** (0.19)	0.84 (0.11)
SMMF	1.017 (0.54)	**1.00** (0.00)	**0.0** (0)	0.08 (0.09)	**0.018** (0.01)	0.08 (0.09)	0.85 (0.09)
Karma	**1.019** (0.54)	0.96 (0.03)	36.5 (10.6)	0.39 (0.18)	0.011 (0.01)	0.41 (0.19)	0.83 (0.11)

Table 2: Simulation results for trace-driven demands.

Nash Welfare (NW) is $\sum_i w_i \log(U_i)$, where $w_i = \frac{e_i}{\sum_j e_j}$. NW balances efficiency and fairness by rewarding high total utility while penalizing inequality. We report NW normalized by the value achieved under the static allocation.

Sharing Index (SIx) for agent i is $\mathrm{SIx}_i = U_i/U_i^{\text{static}}$. It measures the individual gain or loss from sharing; values below one indicate harm. We report the minimum SIx across agents and the percentage of agents with $\mathrm{SIx}_i < 1$.

Weighted Min–Max (WMM) ratio is $\frac{\min_i U_i/w_i}{\max_i U_i/w_i}$. It measures fairness by comparing the worst- and best-off agents after weight normalization, with larger values indicating greater equity. This metric can be sensitive to agents with consistently low demands, for whom U_i/w_i will be small under any mechanism. This motivates the normalized variant below.

Normalized Min–Max (NMM) ratio is $\frac{\min_i \mathrm{SIx}_i}{\max_i \mathrm{SIx}_i}$, capturing inequality in realized performance relative to agents' standalone guarantees.

Weighted Equity (WEq) is $\frac{\min_i U_i/w_i}{\text{median}_i U_i/w_i}$. It compares the worst-off agent to a typical agent after weight normalization,[2] with larger values indicating greater equity. WEq is less sensitive to outliers than the min–max ratios above, though still affected by low-demand agents.

Normalized Equity (NEq) is $\frac{\min_i \mathrm{SIx}_i}{\text{median}_i \mathrm{SIx}_i}$, and is less sensitive than WEq to agents with low demands.

6.3 Results

As shown in Table 2, NW is tightly clustered across mechanisms. Since all four mechanisms are Pareto efficient, their utilitarian welfares $\sum_i U_i$ are identical; thus NW reflects only distributional effects in these simulations. Overall, DMMF and Karma achieve the highest values, with LENDRECOUP extremely close behind, while SMMF trails slightly. The small spread suggests that none of the mechanisms delivers a decisive welfare advantage, and that LENDRECOUP attains near-frontier performance despite its stronger guarantees.

For SIx, LENDRECOUP and SMMF stand out by completely avoiding harm from sharing: both report a minimum value of 1.00 and 0% of agents with $\mathrm{SIx}_i < 1$. In contrast, DMMF and Karma exhibit harm for a substantial minority of agents, with minimum SIx_i values below one (0.96) and roughly a third of agents experiencing losses relative to the constant-allocation baseline ($\approx 36\%$).

These differences matter because SIx_i directly captures whether sharing can leave participants worse off than not participating. By eliminating observed harm (both in worst case and incidence), LENDRECOUP offers a stronger practical value proposition than mechanisms achieving similar welfare while shifting costs onto a subset of agents.

For WMM and WEq, DMMF performs best, with Karma nearly tied, while LENDRECOUP achieves roughly 60% of DMMF's performance; SMMF trails significantly. In contrast, under the normalized metrics NMM and NEq, Karma and DMMF perform relatively worse: SMMF performs best for NMM followed by LENDRECOUP, while LENDRECOUP performs best for NEq. The disparity between the weighted and normalized metrics is notable; understanding the causes of this gap is an interesting direction for future work. Among the mechanisms considered, LENDRECOUP is the most robust across metrics, again suggesting that it yields a more even distribution of utilities across agents.

[2]The inverse of this metric appears in [Vuppalapati *et al.*, 2023], where it is called *disparity*.

7 Conclusion

We introduce the notion of credit fairness in resource-sharing systems, motivated by the idea that agents who lend resources to the system in early rounds should receive priority later. Combined with Pareto efficiency, credit fairness strengthens sharing incentives, making it both a fairness and stability notion for shared systems. Although this strengthening comes at a cost—credit fairness cannot be achieved by an anonymous mechanism simultaneously with both Pareto efficiency and strategyproofness—we introduce LENDRECOUP, a novel mechanism that is credit fair, Pareto efficient, and online strategyproof. Experiments on real-world computational resource-sharing traces show that LENDRECOUP achieves performance comparable to existing state-of-the-art mechanisms. Together with its formal credit-fairness guarantee, these results position LENDRECOUP as a compelling candidate for practical deployment.

Our work raises several open questions. We study a setting with a single resource type, capped linear utility functions, and a fixed set of agents and resource units. Extending credit fairness to more general settings is an interesting direction for future work. Even within our setting, the design space of mechanisms and desiderata remains far from fully explored. For instance, are there mechanisms that achieve stronger theoretical guarantees (e.g., ϵ-Nash SP) and/or improved empirical performance than the four mechanisms studied here?

Acknowledgments

This work was partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grants RGPIN-2019-04936 and DGECR-2019-00475. Both authors contributed equally to this work.

References

[Aleksandrov and Walsh, 2019a] Martin Aleksandrov and Toby Walsh. Monotone and online fair division. In *Proceedings of the Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*, pages 60–75. Springer, 2019.

[Aleksandrov and Walsh, 2019b] Martin Aleksandrov and Toby Walsh. Strategy-proofness, envy-freeness and Pareto efficiency in online fair division with additive utilities. In *Proceedings of the Pacific Rim International Conference on Artificial Intelligence*, pages 527–541, 2019.

[Benadè *et al.*, 2018] Gerdus Benadè, Aleksandr M Kazachkov, Ariel D Procaccia, and Christos-Alexandros Psomas. How to make envy vanish over time. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 593–610, 2018.

[Benadè *et al.*, 2022] Gerdus Benadè, Daniel Halpern, and Alexandros Psomas. Dynamic fair division with partial information. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 3703–3715, 2022.

[Benadè *et al.*, 2024] Gerdus Benadè, Aleksandr M Kazachkov, Ariel D Procaccia, Alexandros Psomas, and David Zeng. Fair and efficient online allocations. *Operations Research*, 72(4):1438–1452, 2024.

[Cookson *et al.*, 2025] Benjamin Cookson, Soroush Ebadian, and Nisarg Shah. Temporal fair division. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 39, pages 13727–13734, 2025.

[Fikioris *et al.*, 2024] Giannis Fikioris, Rachit Agarwal, and Éva Tardos. Incentives in dominant resource fair allocation under dynamic demands. In *Proceedings of the 17th International Symposium on Algorithmic Game Theory*, pages 108–125. Springer, 2024.

[Fikioris *et al.*, 2025] Giannis Fikioris, Siddhartha Banerjee, and Éva Tardos. Beyond worst-case online allocation via dynamic max-min fairness. In *Proceedings of the 26th ACM Conference on Economics and Computation (EC)*, pages 92–92, 2025.

[Freeman *et al.*, 2018] Rupert Freeman, Seyed Majid Zahedi, Vincent Conitzer, and Benjamin C Lee. Dynamic proportional sharing: A game-theoretic approach. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2(1):1–36, 2018.

[Ghodsi *et al.*, 2011] Ali Ghodsi, Matei Zaharia, Benjamin Hindman, Andy Konwinski, Scott Shenker, and Ion Stoica. Dominant resource fairness: Fair allocation of multiple resource types. In *Proceedings of the 8th USENIX Symposium on Networked Systems Design and Implementation*, 2011.

[Gulati *et al.*, 2012] Ajay Gulati, Ganesha Shanmuganathan, Xuechen Zhang, and Peter Varman. Demand based hierarchical {QoS} using storage resource pools. In *Proceedings of the 2012 USENIX Annual Technical Conference*, pages 1–13, 2012.

[Hossain, 2019] Ridi Hossain. Sharing is caring: Dynamic mechanism for shared resource ownership. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2417–2419, 2019.

[Igarashi *et al.*, 2024] Ayumi Igarashi, Martin Lackner, Oliviero Nardi, and Arianna Novaro. Repeated fair allocation of indivisible items. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9781–9789, 2024.

[Kelly *et al.*, 1998] Frank P Kelly, Aman K Maulloo, and David Kim Hong Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49(3):237–252, 1998.

[Li *et al.*, 2021] Fu Li, C Gregory Plaxton, and Vaibhav B Sinha. Egalitarian resource sharing over multiple rounds. *arXiv preprint arXiv:2106.02688*, 2021.

[Reiss *et al.*, 2011] Charles Reiss, John Wilkes, and Joseph L. Hellerstein. Google cluster-usage traces: format + schema. Technical report, Google Inc., Mountain View, CA, USA, November 2011. Revised 2014-11-17 for version 2.1. Posted at https://github.com/google/cluster-data.

[Stoica *et al.*, 2003] Ion Stoica, Scott Shenker, and Hui Zhang. Core-stateless fair queueing: a scalable architecture to approximate fair bandwidth allocations in high-speed networks. *IEEE/ACM Transactions on Networking*, 11(1):33–46, 2003.

[Vardi *et al.*, 2022] Shai Vardi, Alexandros Psomas, and Eric Friedman. Dynamic fair resource division. *Mathematics of Operations Research*, 47(2):945–968, 2022.

[Vuppalapati *et al.*, 2023] Midhul Vuppalapati, Giannis Fikioris, Rachit Agarwal, Asaf Cidon, Anurag Khandelwal, and Éva Tardos. Karma: Resource allocation for dynamic demands. In *Proceedings of the 17th USENIX Symposium on Operating Systems Design and Implementation*, pages 645–662, 2023.

[Wilkes, 2011] John Wilkes. More Google cluster data. Google Research Blog, November 2011. Posted at http://googleresearch.blogspot.com/2011/11/more-google-cluster-data.html.

[Zahedi and Freeman, 2026] Seyed Majid Zahedi and Rupert Freeman. Credit fairness: Online fairness in shared resource pools. *arXiv preprint arXiv:2601.17944*, 2026.

[Zeng and Psomas, 2020] David Zeng and Alexandros Psomas. Fairness-efficiency tradeoffs in dynamic fair division. In *Proceedings of the 21st ACM Conference on Economics and Computation*, pages 911–912, 2020.