Mechanism Design for Efficient Task Allocation

Zifan Gong
City University of Hong Kong
Hong Kong, China
zifan.gong@my.cityu.edu.hk

Minming Li
City University of Hong Kong
Hong Kong, China
minming.li@cityu.edu.hk

Houyu Zhou*
UNSW Sydney
Sydney, Australia
houyu.zhou@unsw.edu.au

ABSTRACT

Task allocation involves a group of agents contributing to various tasks and has been well-studied in resource allocation and related fields. In many scenarios, tasks are distributed across different areas, such as medical jobs in urban and rural regions, and sometimes different tasks require different skill sets. A key challenge is the tendency of agents to choose easier tasks or more prosperous areas for their own benefit. This self-interest can create imbalances, leaving challenging tasks undone or leading to uneven resource distribution, such as the shortage of rural doctors.

To address this problem, we study task allocation using a game-theoretic approach. We model the problem as task allocation games with different tasks and a group of rational, identical agents who strategically select tasks to minimize their workloads. Our goal is to design mechanisms that ensure all workloads are completed in every Nash equilibrium. We show that achieving this requires implementing positive or negative incentives. We then propose effective mechanisms that leverage both types of incentives and extend our results to scenarios with multiple tasks and heterogeneous agents.

KEYWORDS

Nash equilibrium, Task allocation, Mechanism design

ACM Reference Format:
Zifan Gong, Minming Li, and Houyu Zhou. 2026. Mechanism Design for Efficient Task Allocation. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/

1 INTRODUCTION

Many resource allocation problems study how to distribute resources efficiently and fairly. However, in many real-world settings, individuals choose resources themselves, leading to significant societal challenges. For example, in hospitals, nurses and doctors often express shift preferences, but self-selection can result in uneven workloads. Senior staff may avoid night shifts, leaving junior staff to handle undesirable hours, which can lead to burnout and reduced care quality during critical night hours. Likewise, the shortage of rural doctors raises concerns about healthcare access in underserved regions. Urbanization and lifestyle preferences draw healthcare professionals to cities, leaving rural regions understaffed. Similarly, when students freely choose academic majors, many are drawn to popular fields like computer science. This leaves other essential disciplines with too few students, creating an imbalance that reduces the diversity of skills and knowledge in the workforce.

Inspired by these phenomena and the scientific credit allocation problems studied by [13], which aim to promote the completion of research projects, we model our problem as a game-theoretic framework, namely task allocation games. In this setting, individuals select tasks strategically, and the goal is to ensure all tasks are completed. Specifically, we consider several tasks, each with a required workload. A government or organization determines how to allocate workloads, such as dividing them equally among agents choosing the same task. Agents may change their choices to reduce their workload, and the game reaches a Nash equilibrium when no agent has an incentive to deviate. The key question is: What mechanisms can the designer implement to ensure all tasks are completed at every Nash equilibrium?

We first explore whether methods from [13] can address this problem. While their approach achieves social objectives by reallocating workloads among agents and re-weighting task demands, directly applying these methods to our setting is challenging. Consider a simple example with two tasks, A and B, with workloads of 1 and 10, and two agents. Regardless of how the designer allocates workloads, neither agent will choose task B since task A imposes at most a workload of 1. Thus, reallocating workloads alone cannot resolve the issue unless agents are forced to select task B. Re-weighting workloads also presents challenges. Increasing the workload of A may lead to unnecessary work, while reducing the workload of B is often impractical in real-world scenarios. Subsidizing agents to incentivize task selection, as many countries do, offers a potential solution. The key question then is: How can we minimize subsidies while ensuring all tasks are completed?

Our research agenda is structured around the following key questions:

- Is there a task allocation mechanism where every Nash equilibrium ensures that all tasks are completed?
- If we increase the workload of certain tasks, is there a mechanism where every Nash equilibrium guarantees that no unnecessary workload is performed?
- If subsidies are used, how can we minimize them while ensuring that every Nash equilibrium achieves the desired outcome?

1.1 Our Contribution

In this paper, we study task allocation games (TAGs). Our objective is to design mechanisms for TAGs that guarantee the social requirement: ensuring that every Nash equilibrium results in the completion of all tasks. We begin by analyzing TAGs through the

*Corresponding author

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/

lens of mechanism design with subsidy and then generalize the model to include agents with varying valuations and multiple tasks.

- **Agents with Identical Valuations** Our primary contribution is the introduction of a class of mechanisms that strategically subsidize agents. We begin by presenting a specific class of mechanism, SA-1, which subsidizes only one agent and can meet social requirement in multi-task settings. Building on this, we develop a more sophisticated class of mechanisms, SA-X, to minimize the subsidy in scenarios with two tasks. SA-X considers multiple agents and strategically applies subsidies. It first calculates the minimum subsidy required to transition the state into a Nash equilibrium. Subsidies are then carefully assigned and adjusted to ensure that the state requiring the least subsidy becomes the unique Nash equilibrium. The subsidy required for SA-X is significantly smaller than that for SA-1. Furthermore, we improve SA-X and propose CSA-X, a more concise and simpler mechanism.
- **Agents with Different Valuations** Beyond the identical agents setting, we extend our model to scenarios where agents have different valuations for each task, making the problem more challenging. We first consider the case with two tasks. In this scenario, some of the favorable properties observed in the uniform setting no longer hold. To address this, we introduce SA-U, which achieves the minimum subsidy and meets the social requirement in this more complex setting. Finally, we analyze an even more challenging scenario with multiple tasks and diverse agents. In this case, the strategy space grows exponentially, making the direct application of methods from SA-U impractical. To address this, we propose a new mechanism, SA-1D, specifically designed to handle the multi-task scenario effectively.

1.2 Related Work

Since this is a multifaceted problem, we briefly discuss relevant work in multiple directions in the following.

Scientific Credit Allocation. Besides Kleinberg and Oren [12], Bilò et al. [4] introduced a strategic game named the project game, in which each agent can choose a set of projects to work on. The success of each project yields rewards and the reward will be fully allocated among all participating agents. They also proved the existence and computational complexity of Nash equilibrium. However, our model concentrates on designing allocation mechanisms that aim at achieving the social requirement.

Crowdsourcing with Incentives. Crowdsourcing [11] involves breaking a task into smaller pieces and designing mechanisms to attract agents to complete them. Agents then decide whether to participate based on workload and incentives, a process similar to our model. Many studies design incentives in crowdsourcing to attract agents, typically using monetary rewards such as cash or coupons [6, 14].

Resource Games. There are also many studies on allocation with strategic agents [1, 2, 5, 8]. However, in the above works, the action space of the strategic agents is the valuation function, while the action space in our problem is the set of tasks. Additionally, the objectives of these works focus on achieving fairness properties, which is different from our model. Coalitional resource games (CRGs) [7, 15, 16] model cooperative settings where agents, each endowed with resources, can form coalitions to achieve their goals. An agent is satisfied if at least one of its goals is achieved, and agents pool resources to meet the goals. This cooperative structure contrasts with our model, where agents act individually without forming coalitions.

Fair Division with Subsidy. The subsidy is also used in fair division problems in the algorithmic game theory field. Halpern and Shah [10] proved that envy-freeness can be achieved by providing a small amount of money (subsidy), which can be allocated with the indivisible goods. Goko et al. [9] proposed a truthful allocation mechanism for indivisible goods that achieved fairness and efficiency criteria with a limited amount of subsidy. Aziz et al. [3] studied envy-free allocation with subsidy for weighted agents.

2 PRELIMINARIES

We first develop a model with two tasks. Let $N = \{1, \dots, n\}$ be a set of agents. Let $S = \{1, \dots, m\}$ be the set of m tasks, and each corresponds to a workload p_j where $j \in S$. Let $P = \{p_1, \dots, p_m\}$, $p_i > 0$ for $i \in S$ be the task profile, i.e., the workload of each task. Each agent has a strategy $s_i \in S$, corresponding to the task they will select. We denote the strategies of all agents by a strategy profile $\mathbf{s} = \{s_1, \dots, s_n\}$ and denote the strategies of all agents except i by $\mathbf{s}_{-i}$. We use $K_j(\mathbf{s})$ to denote the number of agents with $s_i = j$ in strategy profile $\mathbf{s}$. An allocation mechanism $\mathcal{M}$ outputs an allocation of tasks, given a task profile P and a strategy profile $\mathbf{s}$. We consider three types of allocation mechanisms.

Pure Allocation (PA). *Given a task profile P and a strategy profile $\mathbf{s}$, $\mathcal{M}$ outputs an allocation $A = \{a_1, \dots, a_n\}$ where a_i is the amount of workloads allocated to agent i.*

Allocation with Dummy (DA). *Given a task profile P and a strategy profile $\mathbf{s}$, $\mathcal{M}$ outputs a dummy task profile $D = \{d_1, \dots, d_m\}$, an allocation $A = \{a_1, \dots, a_n\}$ and a dummy allocation $a^+ = \{a_1^+, \dots, a_n^+\}$ where d_j is the amount of dummy workloads introduced to task j, a_i is the amount of workloads allocated to agent i, a_i^+ is the amount of dummy workloads allocated to agent i.*

Subsidy Allocation (SA). *Given a task profile P and a strategy profile $\mathbf{s}$, $\mathcal{M}$ outputs an allocation $A = \{a_1, \dots, a_n\}$ and a subsidy allocation $a^- = \{a_1^-, \dots, a_n^-\}$ where a_i is the amount of workloads allocated to agent i, and a_i^- is the amount of subsidy allocated to agent i.*

Given an instance of task allocation games $(N, P, \mathcal{M})$, the workload of agent $i \in N$ is defined as

$$c_i(\mathcal{M}(\mathbf{s}, P)) = \begin{cases} a_i & \mathcal{M} \in PA \\ a_i + a_i^+ & \mathcal{M} \in DA \\ a_i - a_i^- & \mathcal{M} \in SA \end{cases}.$$

Each agent is rational and aims to minimize their workloads. Figure 1 shows three examples to illustrate how each type of mechanism works in TAG with two agents.

Our goal is designing allocation mechanisms $\mathcal{M}^*$ so that in each task allocation game $(N, P, \mathcal{M}^*)$, every Nash equilibrium can achieve the social requirement, i.e., all tasks can be completed.

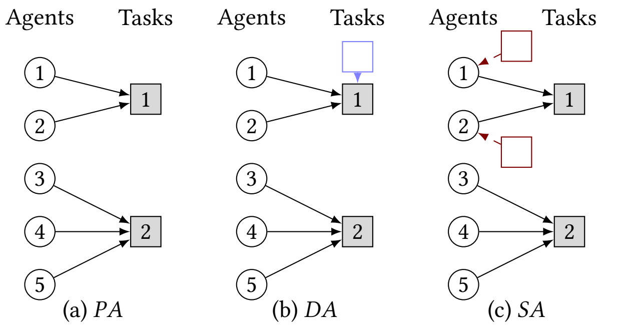

Figure 1: There are two tasks with the workloads 1 each. Suppose that two agents select task 1, and three agents select task 2. (a): PA allocates the workloads to each agent equally. Agents 1-2 are allocated the amount of workloads 1/2 and agents 3-5 are allocated the amount of workloads 1/3. In this instance, agent $1-2$ will deviate and it is not a Nash Equilibrium. (b): DA adds 1/2 dummy workloads to task 1 and allocates the workloads and the dummy workloads to each agent equally. Agents 1-2 have the amount of workloads 3/4 and agents 3-5 have the amount of workloads 1/3. In this instance, agent 1-2 will deviate and it is not a Nash Equilibrium. (c): SA allocates the workloads to each agent equally and only gives agents 1-2 a subsidy of 1/4 each. Agents 1-2 have the amount of workloads 1/4 and agents 3-5 have the amount of workloads 1/3. In this case, no agent will deviate and it achieves Nash Equilibrium.

Definition 1 (Nash Equilibrium (NE)). *We say a strategy $\mathbf{s}$ is a (pure) Nash equilibrium (NE) if no agent can reduce their workload by unilaterally changing their strategy. Formally, given a task allocation game $(N, P, \mathcal{M})$, we say $\mathbf{s}$ is a Nash equilibrium if for all agents $i \in N$, we have*

$$c_i(\mathcal{M}(\{s_i, \mathbf{s}_{-i}\}, P)) \le c_i(\mathcal{M}(\{s_i', \mathbf{s}_{-i}\}, P)),$$

for all $s_i' \in S$, for all P, for all s_{-i}.

Definition 2 (Social Requirement). *Given $(N, P, \mathcal{M})$, we say an NE $\mathbf{s}$ satisfies the social requirement if $\sum_{i:s_i=j} a_i = p_j$ for all $j \in S$.*

2.1 Warm-up

In this subsection, we will discuss PA and DA. First, we show that social requirement cannot be achieved by using PA.

Proposition 1. *There exists a task profile in which for any mechanisms belonging to PA, there is a Nash equilibrium which cannot guarantee the social requirement.*

Proof. Consider a task allocation game $(N, P, \mathcal{M})$ with two tasks, where $P = \{p_1, np_1 + \epsilon\}$. We assume for contradiction that there is $\mathcal{M} \in PA$ under which every NE $\mathbf{s}$ can guarantee the social requirement. Then there exists an agent i with $s_i = 2$ and the workload

$$c_i(\mathcal{M}(\mathbf{s}, P)) \ge \frac{np_1 + \epsilon}{n} > p_1.$$

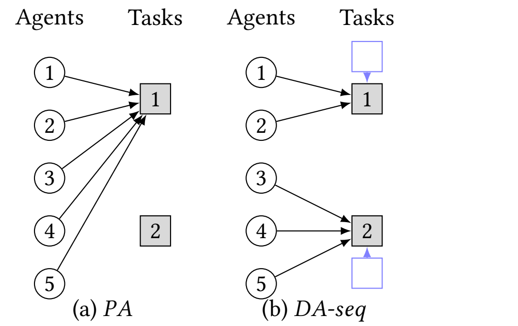

Figure 2: An example to show the NE under different allocation mechanisms.

However, they can change their strategy to $s_i' = 1$ to reduce workload to $c_i(\mathcal{M}(\{s_i' = 1, \mathbf{s}_{-i}\}, P)) \le p_1$, in contradiction to NE. □

Therefore, we need to leverage dummy workload to prevent some agents from selecting the task with lighter workload or subsidy to encourage some agents to select the task with the heavier workload.

Dummy Allocation with sequencing (DA-seq). *Given a task profile P and a strategy profile $\mathbf{s}$, add dummy workload with the amount of d_j to each task j. Set m integers $n_1, \ldots, n_m$, and a strict order $\pi(N)$ on the agents. For each task j, if there are less than n_j agents, allocate the whole original task j to them equally. Otherwise, allocate the whole original task j to n_j agents who are prior to at least $K_j(\mathbf{s}) - n_j$ agents with $s_i = j$ in the order, and allocate the whole dummy workload d_j to the remaining agents with $s_i = j$ equally.*

Intuitively, DA-seq will allocate the dummy workloads to some agents if there are many agents selecting the same task. Note that, the social requirement implies all workloads from original tasks should be allocated. There is no requirement about the dummy workloads since they are fictional or meaningless. Here we give an example of DA-seq to illustrate how it changes the NE.

Example 1. *Given a task profile $P = \{1, 6\}$ and $n = 5$, consider a DA-seq where $d_1 = d_2 = 100$, $\pi(N) = (1, 2, 3, 4, 5)$, $n_1 = 2$, and $n_2 = 3$. We compare it with a PA which allocates the workloads to the respective agents equally. As Figure 2 shows, there is a unique Nash equilibrium under PA: all agents have the same strategy, $s_i = 1$. If we use DA-seq instead of PA, agents 3, 4, 5 will be allocated to the dummy workloads since agents 1, 2 are prior to them in $\pi(N)$. Hence, they will change their strategies to $s_i = 2$. In addition, we can observe that the strategy profile $\mathbf{s} = \{1, 1, 2, 2, 2\}$ is the unique Nash equilibrium under DA-seq.*

If we set all d_j to be sufficiently large, i.e., an agent will reduce workload if they can be allocated the original task instead of the dummy workloads by changing their strategy, we can make every Nash equilibrium satisfy the social requirement by carefully deciding the number n_j.

Proposition 2. *For all TAG with DA-seq, every Nash equilibrium satisfies the social requirement if*

- $n_j \geq 1$ *for all* $j \in S$;
- $\sum_{j \in S} n_j - n_{j'} < n$ *for all* $j' \in S$;
- $d_j > \max_{j' \in S}\{p_{j'}\}(n - n_j)$ *for all* $j, j' \in S$.

Proof. Assume by contradiction that there is an NE in which no agent has the strategy $s_i = j^*$. Since $\sum_{j \in S} n_j - n_{j^*} < n$, there exists an agent with $s_i = j'$ who is allocated the dummy workload, and their workload is at least $\frac{d_{j'}}{n - n_{j'}}$. In this case, they will be allocated the original task by changing their strategy to $s_i = j^*$ and reducing workload (from $\frac{d_{j'}}{n - n_{j'}}$ to p_{j^*}) since $d_{j'} > p_{j^*}(n - n_{j'})$ and $n_{j^*} \geq 1$, in contradiction to NE. □

One may worry that DA-seq will waste resources since agents can be allocated the dummy workloads. In fact, we can see that no one will be allocated the dummy workloads in any Nash equilibrium if we strengthen the second condition proposed in Proposition 2 to $\sum_{j \in S} n_j = n$. In addition, we can also adjust the number of agents in each task by changing n_j, even if we do not force changes to the agents' strategies. While there are some practical examples involving dummy tasks (e.g., hospital administrators can add 'dummy tasks' if too many doctors choose the day shift to prevent night shifts from being understaffed), we recognize that DA-seq may be difficult to apply in some other real-life scenarios because 1) it relies on a predetermined order of agents, which leads to unfairness and 2) the agents always resent working under mechanisms with negative incentives. Due to the simplicity and ease of understanding of the mechanism, we used DA-seq as a warm-up in this paper.

2.2 Paper Organization

In this paper, we study task allocation games (TAGs). Our objective is to design mechanisms for TAGs that guarantee the social requirement: ensuring that every Nash equilibrium results in the completion of all tasks. We begin by analyzing TAGs with mechanisms with subsidy and then generalize the model to include agents with varying valuations and multiple tasks.

In Section 3, we propose a class of mechanisms that incorporate subsidies. In Section 4, we extend the model to account for agents with diverse valuations, focusing on a two-task setting in Section 4.1. Subsequently, in Section 4.2, we examine a more general model involving multiple tasks with diverse agents. Finally, in Section 5, we conclude the paper and propose several directions for future research. Due to the space limit, some proofs are omitted.

3 MECHANISM DESIGN WITH SUBSIDY

The intuitive explanation of SA is that we give some subsidies to the agents to make sure that every task will be selected by some agents. Thus, social requirement can be satisfied. However, different from DA-seq where the dummy workloads will not be allocated in any Nash equilibrium if we carefully select the parameters n_j, we need to give subsidies we announced in the mechanisms when we use SA. Therefore, how to use less subsidies to make every Nash equilibrium meet social requirement is a major challenge.

3.1 SA-1 with Multiple Tasks

A straightforward method is to give one agent some subsidies making him willing to change his strategy to selecting a task that no one selected.

Subsidy Allocation with the First Agent (SA-1). *Given a task profile P and a strategy profile* $\mathbf{s}$, *output an allocation A satisfying* $\sum_{i:s_i=j} a_i = p_j$. *Give the subsidy with the amount of* $(p_j - p_{\min}/n + \epsilon)$ *to the agent with* $s_i = j$ *if* $K_{(j)}(\mathbf{s}) = 1$, *where* $p_{\min} = \min_{j \in S}\{p_j\}$ *and* $K_{(j)}(\mathbf{s})$ *denotes the number of agents choose task* j.

Theorem 1. *For every TAG under SA-1, Nash equilibrium always exists and every Nash equilibrium satisfies the social requirement.*

Proof. Given any TAG with SA-1, $(N, P, SA\text{-}1)$, we first show the existence of Nash equilibrium by showing that $n + 1 - m$ agents select $p_{\min}$ and all the other $m - 1$ agents select the remaining $m - 1$ tasks, respectively. If $n + 1 - m = 1$, the cost of the agent selecting $p_{\min}$ is

$$c_i(SA\text{-}1(P, \mathbf{s})) = p_{\min} - (p_{\min} - \frac{p_{\min}}{n} + \epsilon) = \frac{p_{\min}}{n} - \epsilon.$$

If they select another task, the cost will be increased from $\frac{p_{\min}}{n} - \epsilon$ to at least $\frac{p_{\min}}{2}$. Therefore, they will not change their strategy.

If $n + 1 - m > 1$, the cost of the agent selecting $p_{\min}$ is

$$c_i(SA\text{-}1(P, \mathbf{s})) = \frac{p_{\min}}{n + 1 - m}.$$

If they select another task, the cost will be increased from $\frac{p_{\min}}{n+1-m}$ to at least $\frac{p_{\min}}{2}$. Therefore, they will not change their strategy.

Then we consider the other agents. The cost of agent selecting p_j is

$$c_i(SA\text{-}1(P, \mathbf{s})) = p_j - (p_j - \frac{p_{\min}}{n} + \epsilon) = \frac{p_{\min}}{n} - \epsilon.$$

If they select another task, the cost will be increased from $\frac{p_{\min}}{n} - \epsilon$ to at least $\frac{p_{\min}}{n+2-m}$. Therefore, Nash equilibrium always exists.

Then we prove that every Nash equilibrium guarantees the social requirement. We show that $K_j(\mathbf{s}) > 0$ for all $j \in S$ and for every Nash equilibrium $\mathbf{s}$. Assume by contradiction that $K_j(\mathbf{s}) = 0$. There is an agent i with strategy $s_i = j'$ whose workload is

$$c_i(SA\text{-}1(P, \mathbf{s})) = \frac{p_{j'}}{K_{(j')}(\mathbf{s})} \geq \frac{p_{\min}}{n}.$$

If they changes their strategy to $s_i = j$, they will be the only agent who selects j and the workload will be

$$p_j - (p_j - \frac{p_{\min}}{n} + \epsilon) = \frac{p_{\min}}{n} - \epsilon.$$

Therefore they will change their strategy, in contradiction to NE. □

3.2 SA-X with Two Tasks

It seems that $\max_{j \in S}(p_j - p_{\min}/n + \epsilon)$ is the minimum subsidy we have to use to make every Nash equilibrium satisfy the social requirement since less subsidy would cause everyone to select the easier task and no one to select the other. However, if we continue to commit to subsidizing more agents to entice them to change their strategy to selecting the heavier task until we do not need so many subsidies in total to maintain the newly produced Nash equilibrium, the real subsidies required will decrease. Consider the following example.

Example 2. *Assume that $p_1 = p_2 = 1$ and $n = 4$, we allocate the same amount of workloads to each agent on the corresponding task. In every Nash equilibrium, all agents have the same strategy and each has workload $\frac{1}{4}$. Without loss of generality, we assume that all agents have the strategy $s_i = 1$.*

First, we promise agent 1 a subsidy of $\frac{3}{4} + \epsilon$ if they can change their strategy to $s'_1 = 2$. Then agent 1 is willing to change since their workload will decrease.

Next, we promise agents 1 and 2 a subsidy of $\frac{1}{6} + \epsilon$ per agent if agent 2 can change their strategy to $s'_2 = 2$. Agent 1 is still willing to stick to task 2 even with fewer subsidies to them since they will increase workload (from $\frac{1}{2} - (\frac{1}{6} + \epsilon)$ to $\frac{1}{3}$) by changing their strategy if agent 2 has changed their strategy to $s'_2 = 2$. On the other hand, agent 2 is willing to change their strategy to $s'_2 = 2$ since it reduces their workload from $\frac{1}{3}$ to $\frac{1}{2} - (\frac{1}{6} + \epsilon)$.

Finally, we promise the other two agents a subsidy of $\frac{1}{6} + \epsilon$ each to make them still select task 1. Although constructing that new Nash equilibrium requires subsidizing all agents, the amount of the total subsidy needed is $4 \times (\frac{1}{6} + \epsilon) < \frac{3}{4} + \epsilon$.

One may ask whether we can skip the first step (promising agent 1 a subsidy of $\frac{3}{4} + \epsilon$). If we remove that step, every strategy profile where all agents select task 1 is still a Nash equilibrium since every single agent cannot reduce the workload by changing their strategy. Therefore, no agent will choose task 2, which does not satisfy the social requirement. Hence, all steps are necessary. Then we start with two agents and formalize the above idea as our new allocation mechanism.

Subsidy Allocation with a series of agents (SA-X). *Given a task profile P and a strategy profile $\mathbf{s}$, output an allocation A where $\sum_{i:s_i=j} a_i = p_j$ and $a_i = \frac{p_{s_i}}{K_{s_i}(\mathbf{s})}$. Give the subsidy of C_{jk} to every agent with $s_i = j$ if $K_j(\mathbf{s}) = k$ where $j \in S$. For better understanding and to avoid ambiguity, C_{jk} represents the subsidy required when exactly k agents choose task j. C_{jk} is decided in the following way:*

Initializing: $C_{jn} \Leftarrow 0$ *where* $j \in S$;

$$C_{jk} \Leftarrow \begin{cases} \frac{p_j}{k} - \frac{p_{3-j}}{n-k+1} + \epsilon & \frac{p_j}{k} > \frac{p_{3-j}}{n-k+1}; \\ 0 & otherwise. \end{cases}$$

where $j \in S$, $k \in [n-1]$;

$$\textit{Define } S_k \Leftarrow kC_{1k} + (n-k)C_{2(n-k)}$$

where $k \in [n-1]$.

Tuning: *Let* $k^* = \arg\min_{k\in[n]}\{S_k\}$ *and* $\epsilon^+ > \epsilon$

If $C_{2(n-k^*)} = 0$ **:**

$$C_{2(n-k^*)} \Leftarrow C_{2(n-k^*)} + \epsilon^+;$$

If $C_{1k^*} = 0$ **:**

$$C_{1k^*} \Leftarrow C_{1k^*} + \epsilon^+;$$

Otherwise: *do not tune.*

End

Given a 2-task allocation game with SA-X $(N, P, SA\text{-}X)$, all agents with the same strategy have the same workload (the same amount of workload and the same amount of subsidy). Thus, we denote $(k, n-k)$ as a class of strategy profile $\mathbf{s}$ with $K_1(\mathbf{s}) = k$ and $K_2(\mathbf{s}) = n-k$. Here we reuse the profile given in Example 2 to show how SA-X works, and Table 1 shows the workload of each agent under different strategy profiles. Note that $k^* = 2$ in that example, and we do not need to tune the subsidy. After giving the subsidy, $(2, 2)$ is the only NE in the example. However, not all games can skip the tuning step. Example 3 presents a game where the tuning is needed.

	$s_i = 1$	$s_i = 2$	S_k
$(0, 4)$		$\frac{1}{4}$	
$(1, 3)$	$1 - (\frac{3}{4} + \epsilon)$	$\frac{1}{3} - 0$	$\frac{3}{4} + \epsilon$
$(2, 2)$	$\frac{1}{2} - (\frac{1}{6} + \epsilon)$	$\frac{1}{2} - (\frac{1}{6} + \epsilon)$	$\frac{2}{3} + 4\epsilon$
$(3, 1)$	$\frac{1}{3} - 0$	$1 - (\frac{3}{4} + \epsilon)$	$\frac{3}{4} + \epsilon$
$(4, 0)$	$\frac{1}{4}$		

Table 1: The workload of each agent in Example 2 using SA-X. $(2, 2)$ **is the only NE and the corresponding subsidy is** $\frac{2}{3} + 4\epsilon$**. The red part in brackets corresponds to the subsidy, e.g.,** $C_{11} = \frac{3}{4} + \epsilon$.

Example 3. *Consider a profile with $p_1 = 5$, $p_2 = 2$, and $n = 10$. We use SA-X and the result is shown in Table 2, both (0,10) and (10,0) are Nash equilibria after allocating the task. There is only one NE (7,3) after the initializing process in SA-X. There is only one NE (8,2) after tuning, which uses the minimum subsidy among all strategy profiles.*

The intuitive explanation of deciding the subsidy in SA-X is that: we first decide how to make a strategy profile stable unilaterally by using the minimum amount of subsidy (S_k). After applying the calculated subsidies to all strategy profiles, the new NE is produced. Then we need to find which strategy profile uses the minimum subsidy (S_{k^*}). If $(k^*, n-k^*)$ is the only NE, we do nothing. Otherwise, we increase the subsidy a little (ϵ^+) on $(k^*, n-k^*)$ to make it the only NE.

From the definition of SA-X, we can see that $(k^*, n-k^*)$ uses the minimum subsidy among all strategy profiles. Next, we need to show that $(k^*, n-k^*)$ is the only NE in the 2-task allocation game with SA-X. We first characterize all NE in 2-TAG under the mechanism which allocates the tasks to the respective agents equally.

Pure Allocation with Equalization (PA-avg). *Given a task profile P and a strategy profile $\mathbf{s}$, output an allocation A where $\sum_{i:s_i=j} a_i = p_j$ and $a_i = \frac{p_{s_i}}{K_{s_i}(\mathbf{s})}$.*

Proposition 3. *Given any 2-task allocation game with PA-avg, $s_1 = \cdots = s_n$ in every Nash equilibrium $\mathbf{s}$.*

Proof. We prove this by making a contradiction. Suppose there is a Nash equilibrium $\mathbf{s}$ with $K_j(\mathbf{s}) > 0$ where $j = 1, 2$. The workload of agent i is equal to $\frac{p_j}{K_j(\mathbf{s})}$ where $s_i = j$. Since $\mathbf{s}$ is Nash equilibrium,

	Initializing		Tuning		
	$s_i = 1$	$s_i = 2$	$s_i = 1$	$s_i = 2$	S_k
$(0, 10)$		$\frac{1}{5}$		$\frac{1}{5}$	
$(1, 9)$	$5 - (\frac{24}{5} + \epsilon)$	$\frac{2}{9} - 0$	$\frac{1}{5} - \epsilon$	$\frac{2}{9} - 0$	$\frac{24}{5} + \epsilon$
$(2, 8)$	$\frac{5}{2} - (\frac{41}{18} + \epsilon)$	$\frac{1}{4} - 0$	$\frac{2}{9} - \epsilon$	$\frac{1}{4} - 0$	$\frac{41}{9} + 2\epsilon$
$(3, 7)$	$\frac{5}{3} - (\frac{17}{12} + \epsilon)$	$\frac{2}{7} - 0$	$\frac{1}{4} - \epsilon$	$\frac{2}{7} - 0$	$\frac{17}{4} + 3\epsilon$
$(4, 6)$	$\frac{5}{4} - (\frac{27}{28} + \epsilon)$	$\frac{1}{3} - 0$	$\frac{2}{7} - \epsilon$	$\frac{1}{3} - 0$	$\frac{27}{7} + 4\epsilon$
$(5, 5)$	$1 - (\frac{2}{3} + \epsilon)$	$\frac{2}{5} - 0$	$\frac{1}{3} - \epsilon$	$\frac{2}{5} - 0$	$\frac{10}{3} + 5\epsilon$
$(6, 4)$	$\frac{5}{6} - (\frac{13}{30} + \epsilon)$	$\frac{1}{2} - 0$	$\frac{2}{5} - \epsilon$	$\frac{1}{2} - 0$	$\frac{13}{5} + 6\epsilon$
$(7, 3)$	$\frac{5}{7} - (\frac{3}{14} + \epsilon)$	$\frac{2}{3} - (\frac{1}{24} + \epsilon)$	$\frac{1}{2} - \epsilon$	$\frac{5}{8} - \epsilon$	$\frac{13}{8} + 10\epsilon$
$(8, 2)$	$\frac{5}{8} - 0$	$1 - (\frac{4}{9} + \epsilon)$	$\frac{5}{8} - \epsilon^+$	$\frac{5}{9} - \epsilon$	$\frac{8}{9} + 2\epsilon$
$(9, 1)$	$\frac{5}{9} - 0$	$2 - (\frac{3}{2} + \epsilon)$	$\frac{5}{9}$	$\frac{1}{2} - \epsilon$	$\frac{3}{2} + \epsilon$
$(10, 0)$	$\frac{1}{2}$		$\frac{1}{2}$		

Table 2: The workload of each agent in Example 3 by using SA-X. $(8, 2)$ is the only NE and the corresponding subsidy is $\frac{8}{9} + 2\epsilon + \epsilon^+$. The red part in brackets corresponds to the subsidy, e.g., $C_{11} = \frac{24}{5} + \epsilon$.

no agent can reduce workload by unilaterally changing their strategy. Thus, we have $\frac{p_j}{K_j(\mathbf{s})} \leq \frac{p_{3-j}}{K_{3-j}(\mathbf{s})+1}$. Then from $\frac{p_j}{K_j(\mathbf{s})+1} < \frac{p_j}{K_j(\mathbf{s})}$ we further have $\frac{p_j}{K_j(\mathbf{s})} < \frac{p_{3-j}}{K_{3-j}(\mathbf{s})}$, i.e., $\frac{p_1}{K_1(\mathbf{s})} < \frac{p_2}{K_2(\mathbf{s})}$ and $\frac{p_2}{K_2(\mathbf{s})} < \frac{p_1}{K_1(\mathbf{s})}$, which makes a contradiction. □

Note that PA-avg is the same as the part of SA-X before allocating the subsidy. Therefore, no strategy profile $(k, n-k)$ where $k \in [n-1]$ is NE without using the subsidy, implying $S_k > 0$. A straightforward question is, can any $k \in [n-1]$ be k^*? We observe that the tuning step is only divided into three cases, thus k^* can take only specific values. Next, we formally show this through a series of propositions. We first show the monotonicity in S_k. Since S_k is independent of the tuning step in SA-X, we consider a simpler SA-X mechanism without the tuning step. Let k_l be the maximum k with $C_{2(n-k)} = 0$ and k_r be the minimum k with $C_{1k} = 0$, we have the following proposition.

Proposition 4. *Given any 2-task allocation game $(N, P, SA\text{-}X)$ without tuning. S_k is decreasing when $1 \leq k \leq k_l$ and increasing when $k_r \leq k \leq n-1$.*

Proof. For $1 \leq k \leq k_l$, we have

$$\frac{p_1}{n-k} - \frac{p_2}{k+1} \leq \frac{p_1}{n-k_l} - \frac{p_2}{k_l+1} \leq 0,$$

implying that $C_{2(n-k)} = 0$ for all $k \in [k_l]$. Therefore,

$$S_k = kC_{1k} = p_1 - k\frac{p_2}{n-k+1} + k\epsilon,$$

which is monotonically decreasing for all $k \in [k_l]$.

We can use the similar analysis to show $C_{1k} = 0$ when $k_r \leq k \leq n-1$ and $S_k = (n-k)C_{2(n-k)}$, which is monotonically increasing. □

From Table 2 we can also observe that S_k decreases when $1 \leq k \leq 6$ and increases when $8 \leq k \leq 9$, which coincides with our proof. Therefore, we have $k_l \leq k^* \leq k_r$. Consider a sequence of strategy profiles $(0, n), (1, n-1), \cdots, (n, 0)$, we observe that $(k_l, n-k_l)$ is the last strategy profile which does not subsidize the agents with strategy $s_i = 2$, and $(k_r, n-k_r)$ is the first strategy profile which does not subsidize the agents with strategy $s_i = 1$, when we use SA-X without tuning. Hence, if we know the number of strategy profiles between k_l and k_r, the search range for k^* will be significantly reduced.

Proposition 5. *Given any 2-task allocation game $(N, P, PA\text{-}avg)$, there is at most one strategy profile $(k, n-k)$ with $k \in [n-1]$, where every agent can reduce workload by unilaterally changing strategy.*

Proof. Given profiles N and P, if there is an agent that cannot reduce workload by changing their strategy for every strategy profile, the proposition holds trivially. For the remaining cases, we show that the strategy profile where every agent can reduce workload by unilaterally changing their strategy is unique. Suppose we have such a strategy profile $\mathbf{s}$. The agents with $s_i = j$ have workloads of $\frac{p_j}{K_j(\mathbf{s})}$. Because every agent can reduce workload by unilaterally changing their strategy, we have

$$\frac{p_j}{K_j(\mathbf{s})} > \frac{p_{3-j}}{K_{3-j}(\mathbf{s})+1}.$$

We further have

$$\frac{p_j}{K_j(\mathbf{s})-k+1} \geq \frac{p_j}{K_j(\mathbf{s})} > \frac{p_{3-j}}{K_{3-j}(\mathbf{s})+1} > \frac{p_{3-j}}{K_{3-j}(\mathbf{s})+k},$$

where $0 < k < K_j(\mathbf{s})$ and $k \in \mathbb{Z}$, implying that any agent with strategy $s_i = 3-j$ cannot reduce workload by changing their strategy to $s_i = j$ for all strategy profiles $(K_j(\mathbf{s}) - k, K_{3-j}(\mathbf{s}) + k)$. Due to the symmetry, any agent with strategy $s_i = j$ cannot reduce workload by changing their strategy to $s_i = 3-j$ for all strategy profiles $(K_j(\mathbf{s}) + k, K_{3-j}(\mathbf{s}) - k)$ where $0 < k < K_{3-j}(\mathbf{s})$. Hence, $\mathbf{s}$ is the only strategy profile where every agent from both tasks can reduce workload by changing their strategy. This completes the proof. □

In SA-X, we subsidize agents to achieve the Nash equilibrium. If an agent is willing to reduce the workload by changing strategy, we will give a subsidy. Therefore, Proposition 5 also implies that there is at most one strategy profile that needs to subsidize both agents from tasks 1 and 2. From Table 2 we observe that $(7, 3)$ is the only strategy profile that subsidizes all agents with $s_i = 1$ and $s_i = 2$, the strategy profiles above $(7, 3)$ only subsidize agents with $s_i = 1$, and the strategy profiles below $(7, 3)$ only subsidize agents with $s_i = 2$, consistent with our result.

Corollary 1. *Given any 2-task allocation game $(N, P, SA\text{-}X)$ without tuning, there is at most one strategy profile $(k, n-k)$ where $k \in [n-1]$, which needs to subsidize every agent with strategies $s_i = 1$ and $s_i = 2$.*

Finally, given a 2-task allocation game $(N, P, SA\text{-}X)$, we have S_{k^*} achieved by one of at most three strategy profiles. To facilitate the description, we denote all strategy profiles $(k, n-k)$ where $\max\{k_l, 1\} \leq k \leq \min\{k_r, n-1\}$ as **undemanding strategy profiles** (here we exclude $k = 0$ and $k = n$ since the domain of S_k

is $[n-1]$). In the following, we first show that every NE belongs to one of the undemanding strategy profiles when we use SA-X without tuning. Then we show that $(k^*, n-k^*)$ is the only NE after tuning.

Proposition 6. *Given any 2-task allocation game $(N, P, SA\text{-}X)$ without tuning, a strategy profile is Nash equilibrium if and only if it belongs to undemanding strategy profiles when $k_r - k_l = 1$; undemanding strategy profile $(\frac{k_l+k_r}{2}, n - \frac{k_l+k_r}{2})$ is the only Nash equilibrium when $k_r - k_l = 2$.*

Theorem 2. *Given any 2-task allocation game $(N, P, SA\text{-}X)$, $(k^*, n-k^*)$ is the only Nash Equilibrium.*

Proof. If $C_{2(n-k^*)} = 0$ and $C_{2(n-k^*-1)} \neq 0$, k^* is the maximum k with $C_{2(n-k)} = 0$. Thus, we have $k^* = k_l \geq 1$. There are two cases satisfying that condition:

(1) $k_r - k_l = 1$ and $k_r \leq n-1$;
(2) $k_r - k_l = 2$.

For both cases, consider strategy profile $(k^*+1, n-k^*-1)$. The workload of any agent with strategy $s_i = 1$ is at least

$$\frac{p_1}{k^*+1} - (\frac{p_1}{k^*+1} - \frac{p_2}{n-k^*} + \epsilon) = \frac{p_2}{n-k^*} - \epsilon$$

If they change their strategy to $s_i' = 2$, the workload will be $\frac{p_2}{n-k^*} - \epsilon^+$, which is less than $\frac{p_2}{n-k_l} - \epsilon$. Thus, $(k^*+1, n-k^*)$ is not NE. It also implies that any agent with strategy $s_i = 2$ cannot reduce workload by changing their strategy to $s_i' = 1$ in strategy profile $(k^*, n-k^*)$. In addition, by Proposition 6, any agent with strategy $s_i = 1$ cannot reduce workload by changing their strategy to $s_i' = 2$ in strategy profile $(k^*, n-k^*)$, and any strategy profile $(k, n-k)$ where $k \leq k^*$ and $k \geq k^*+1$ cannot be NE. Hence, $(k^*, n-k^*)$ is the only NE.

If $C_{1k^*} = 0$ and $C_{1(k^*-1)} \neq 0$, k^* is the minimum k with $C_{1k} = 0$. Thus, we have $k^* = k_r \leq n-1$. There are also two cases satisfying the conditions:

(1) $k_r - k_l = 1$ and $k_l \geq 1$;
(2) $k_r - k_l = 2$.

We can use a similar analysis as above to show that $(k^*, n-k^*)$ is the only NE.

Then we discuss the remaining cases.

If $k^* = \frac{k_l+k_r}{2}$ and $k_r - k_l = 2$, $(\frac{k_l+k_r}{2}, n - \frac{k_l+k_r}{2})$ is the only NE. Hence, we do not need to tune the subsidy. If $k^* = k_l$, $k_r - k_l = 1$, and $k_r = n$, $(k_l, n-k_l)$ is the only NE. Hence, we do need to tune the subsidy. If $k^* = k_r$, $k_r - k_l = 1$, and $k_l = 0$, $(k_r, n-k_r)$ is the only NE. Hence, we do not need to tune the subsidy. This completes the proof. □

3.3 Concave SA-X with Two Tasks

In this subsection, we improve our SA-X by leveraging a concave function to calculate the bonus given in the **Tuning** step, incorporating a suggestion from an anonymous reviewer in our previous submission.

Concave SA-X (CSA-X). *Given a task profile P and a strategy profile $\mathbf{s}$, output an allocation A where $\sum_{i:s_i=j} a_i = p_j$ and $a_i = \frac{p_{s_i}}{K_{s_i}(\mathbf{s})}$. Give the subsidy of $C_j(n_1, n_2)$ to every agent with $s_i = j$ if $K_j(\mathbf{s}) = n_j$ for every $j \in S$, where $C_j(n_1, n_2)$ is decided in the following way:*

Initializing: *for all $j \in S$ and $\sum_{j \in S} n_j = n$.*

$$C_j(n_1, n_2) \Leftarrow 0, \text{ if there exists an } n_j = n;$$
$$C_j(n_1, n_2) \Leftarrow \max(0, \frac{p_j}{n_j} - \frac{p_{3-j}}{n_{3-j}+1}), \text{ otherwise.}$$

$$S(n_1, n_2) \Leftarrow +\infty, \text{ if there exists an } n_j = n;$$
$$S(n_1, n_2) \Leftarrow \sum_{j \in S} n_j C_j(n_1, n_2), \text{ otherwise.}$$

Tuning: *Let $(n_1^*, n_2^*) = \arg\min\{S(n_1, n_2)\}$,*

$$g(n_1, n_2) \Leftarrow n^3 - \sum_{j \in S} (n_j - n_j^*)^2,$$
$$C_j(n_1, n_2) \Leftarrow C_j(n_1, n_2) + g(n_1, n_2)\epsilon$$

In CSA-X mechanism, we introduced the function $g(n_1, n_2)$. Each subsidy C is divided into two components: the original subsidy, given by

$$\max\left(0, \frac{p_j}{n_j} - \frac{p_{3-j}}{n_{3-j}+1}\right),$$

and the bonus, given by $g(k)\epsilon$.

In the tuning process, $g(k)$ is defined as

$$g(n_1, n_2) = n^3 - \sum_{j \in S} (n_j - n_j^*)^2,$$

ensuring that $g(k)$ reaches its maximum at the total-subsidy-minimizer n_j^*. CSA-X ensures two properties.

- Without Tuning, the cost will not change if the agent changes their strategy.
- If an agent changes their strategy in a way that brings the state closer to (n_1^*, n_2^*), then they get more bonus. (The distance from state (n_1, n_2) to (n_1^*, n_2^*) is defined as the Euclidean distance.)

To see the first property, without loss of generality, we assume that the agent at j is going to change their strategy. If $\frac{p_j}{n_j} \geq \frac{p_{3-j}}{n_{3-j}+1}$, their cost will change from $\frac{p_j}{n_j}$ to $\frac{p_{3-j}}{n_{3-j}+1} - (\frac{p_{3-j}}{n_{3-j}+1} - \frac{p_j}{n_j})$, implying the cost will not change. If $\frac{p_j}{n_j} < \frac{p_{3-j}}{n_{3-j}+1}$, their cost will change from $\frac{p_j}{n_j} - (\frac{p_j}{n_j} - \frac{p_{3-j}}{n_{3-j}+1})$ to $\frac{p_{3-j}}{n_{3-j}+1}$, implying the cost will not change. In addition, it is easy to verify that $g(n_1, n_2)$ will be larger if (n_1, n_2) is closer to (n_1^*, n_2^*). Under those properties, all agents have incentive to make the current state closer to (n_1^*, n_2^*), and no agent under the state $(k^*, n-k^*)$ has an incentive to deviate.

However, extending either SA-X or CSA-X to handle multiple tasks faces a great challenge. 1) As the number of tasks expands, the complexity surges due to the exponential growth in the number of the strategy profiles. Consequently, beyond proposing mechanisms to meet the social requirement, implementing these mechanisms in polynomial time emerges as a substantial challenge. In a multi-task scenario, SA-X has to calculate the subsidies for all strategy profiles, which is inevitably time-consuming. 2) Furthermore, the efficacy of SA-X depends on the property of monotonicity, which ensures that the new Nash Equilibrium must belong to one of the undemanding strategy profiles. However, in the presence of multiple tasks, maintaining this monotonicity becomes increasingly challenging.

4 AGENTS WITH DIFFERENT VALUATIONS

In this section, we extend the model so that each agent has different valuations toward tasks (i.e., diverse agents), capturing differences in abilities or preferences of agents, e.g., the agent who is good at doing task A will have a lower workload on task A. We start with 2-task allocation games.

4.1 2-Task Allocation Games

Let $N = \{1, \ldots, n\}$ be the set of agents and $P = \{p_1, p_2\}$ be the task profile. Each agent i has a valuation function $v_i = \{\frac{1}{p_{i1}}, \frac{1}{p_{i2}}\}$. Agent i's cost is $\frac{p_{ij}}{K_j(\mathbf{s})}$ if he chooses task j since higher valuation means higher ability and therefore lower cost.

Theorem 3. *For any 2-task allocation game with diverse agents under SA-1, every Nash equilibrium satisfies the social requirement.*

While SA-1 can always satisfy the social requirement with diverse agents, the SA-X may not achieve Nash equilibrium with the minimum subsidy, because if the agents have different workload profiles, the monotonicity property of total subsidy does not hold. The tuning process in SA-X depends on the monotonicity property, which ensures that the new Nash Equilibrium must belong to one of the undemanding strategy profiles. Moreover, if the agents are different, the subsidy for each agent also varies, making the setting more challenging. Directly extending CSA-X to this setting is difficult, as some properties no longer hold, complicating the computation and adjustment of subsidies. However, it is feasible to extend SA-X by adapting its structure to treat each agent individually. Therefore, we propose the new mechanism SA-U.

Subsidy Allocation with Undemanding agents (SA-U) . *Given a task profile P, a strategy profile $\mathbf{s}$, output an allocation A where $\sum_{i:s_i=j} a_i = p_j$ and $a_i = p_{s_i}/K_{s_i}(\mathbf{s})$. Give the subsidy of C^i_{jk} to agent i with $s_i = j$ if $K_j(\mathbf{s}) = k$ where $j \in S$, where C^i_{jk} is decided in the following way:*

Initializing: $C^i_{jn} \Leftarrow 0$ *where* $j \in S$;

$$C^i_{jk} \Leftarrow \begin{cases} \frac{p_{ij}}{k} - \frac{p_{i3-j}}{n-k+1} + \epsilon & \frac{p_{ij}}{k} > \frac{p_{i3-j}}{n-k+1}; \\ 0 & otherwise. \end{cases}$$

Selecting: *Sort* C^i_{jk} *in ascending order,*

$$\mathbf{C}_N = \left\{C^i_{jk} \,\middle|\, i,k \text{ are distinct}\right\}, |\mathbf{C}_N| = N.$$

$$\forall j,k : C^i_{jk} \Leftarrow \begin{cases} C^i_{jk} & \text{if } C^i_{jk} \in \mathbf{C}_N \\ 0 & \text{otherwise} \end{cases}$$

$$\text{Define } S_k \Leftarrow \sum_{i,s_i=1}^{n} C^i_{1k} + \sum_{i,s_i=2}^{n} C^i_{2(n-k)}$$

where $k \in [n-1]$.

Let $\mathbf{K}^{ne}$ *denote the set of* k *in NE* $(k, n-k)$.

Tuning: *Let* $k^* = \arg\min_{k \in [n]} \{S_k\}$

If $p_{i1}/(k^*+1) > p_{i2}/(n-k^*)$ **:**

$$C^i_{1k} \Leftarrow 0, k \in [k^*+1, n-1]$$

Otherwise:

$$C^i_{2k} \Leftarrow 0, k \in [n-k^*+1, n-1]$$

$$S_k \Leftarrow \sum_{i,s_i=1}^{n} C^i_{1k} + \sum_{i,s_i=2}^{n} C^i_{2(n-k)}$$

where $k \in [n-1]$.

Theorem 4. *For any 2-task allocation game with diverse agents under SA-U, there is only one Nash equilibrium $(k^*, n-k^*)$ with the minimum subsidy.*

4.2 Multi-Task Allocation Games

In this subsection, we further extend the model to multiple tasks as well as diverse agents. Formally, there are m tasks $P = \{p_1, p_2, \ldots p_m\}$. Let $N = \{1, \ldots, n\}$ be the set of agents. Each agent i has a valuation profile $v_i = \{\frac{1}{p_{i1}}, \frac{1}{p_{i2}}, \ldots, \frac{1}{p_{im}}\}$ and p_{ij} stands for agent i's cost toward task j, $j \in S$. We propose a new mechanism for multiple tasks and diverse agents.

SA-1 with Diverse Agents (SA-1D). *Given a task profile P and a strategy profile $\mathbf{s}$, output an allocation A satisfying $\sum_{i:s_i=j} a_i = p_j$. For each agent i, sort p_{ij} in ascending order such that $p_{i1} < p_{i2} < \cdots < p_{im}$. Let*

$$C^i_j = p_{ij} - \frac{p_{i1}}{n-1} + \epsilon.$$

Give the subsidy of C^i_j to the agent if $s_i = j$ and $K_{(j)}(\mathbf{s}) = 1$, where $j \in S$.

Theorem 5. *For any task allocation game with multiple tasks and different agent values, SA-1D always achieves Nash equilibrium and satisfies the social requirement.*

5 DISCUSSION AND CONCLUSION

We study task allocation games where each agent wants to minimize their workload by selecting the task strategically. We propose several allocation mechanisms to make sure that every NE can guarantee that all tasks are assigned to agents.

While our model is quite general, it could be extended further to capture more real-life scenarios. In particular, one can consider a case in which those tasks cannot be divided arbitrarily, that is, indivisible resource allocation. Therefore, how to guarantee fairness between agents when the workload cannot be divided into equal parts is an interesting problem. One can also investigate what happens if agents have mixed strategies rather than pure strategies or limited strategy space.

ACKNOWLEDGMENTS

Minming Li is supported by a grant from Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 11216725). Houyu Zhou is supported by the ARC Laureate Project FL200100204 on "Trustworthy AI".

REFERENCES

[1] Georgios Amanatidis, Georgios Birmpas, George Christodoulou, and Evangelos Markakis. 2017. Truthful allocation mechanisms without payments: Characterization and implications on fairness. In *Proceedings of the 2017 ACM Conference on Economics and Computation*. 545–562.

[2] Georgios Amanatidis, Georgios Birmpas, Federico Fusco, Philip Lazos, Stefano Leonardi, and Rebecca Reiffenhäuser. 2021. Allocating Indivisible Goods to Strategic Agents: Pure Nash Equilibria and Fairness. In *International Conference on Web and Internet Economics*. 149–166.

[3] Haris Aziz, Xin Huang, Kei Kimura, Indrajit Saha, Zhaohong Sun, Mashbat Suzuki, and Makoto Yokoo. 2024. Weighted Envy-free Allocation with Subsidy. *arXiv preprint arXiv:2408.08711* (2024).

[4] Vittorio Bilò, Laurent Gourvès, and Jérôme Monnot. 2019. Project games. In *Algorithms and Complexity: 11th International Conference, CIAC 2019, Rome, Italy, May 27–29, 2019, Proceedings 11*. Springer, 75–86.

[5] Ioannis Caragiannis, Christos Kaklamanis, Panagiotis Kanellopoulos, and Maria Kyropoulou. 2009. On Low-Envy Truthful Allocations. In *International Conference on Algorithmic Decision Theory*. 111–119.

[6] Dominic DiPalantino and Milan Vojnovic. 2009. Crowdsourcing and all-pay auctions. In *Proceedings of the 10th ACM conference on Electronic commerce*. 119–128.

[7] Paul E Dunne, Sarit Kraus, Efrat Manisterski, and Michael Wooldridge. 2010. Solving coalitional resource games. *Artificial Intelligence* 174, 1 (2010), 20–50.

[8] Farzaneh Farhadi, S Jamaloddin Golestani, and Demosthenis Teneketzis. 2018. A surrogate optimization-based mechanism for resource allocation and routing in networks with strategic agents. *IEEE Trans. Automat. Control* 64, 2 (2018), 464–479.

[9] Hiromichi Goko, Ayumi Igarashi, Yasushi Kawase, Kazuhisa Makino, Hanna Sumita, Akihisa Tamura, Yu Yokoi, and Makoto Yokoo. 2024. A fair and truthful mechanism with limited subsidy. *Games and Economic Behavior* 144 (2024), 49–70.

[10] Daniel Halpern and Nisarg Shah. 2019. Fair division with subsidy. In *Algorithmic Game Theory: 12th International Symposium, SAGT 2019, Athens, Greece, September 30–October 3, 2019, Proceedings 12*. Springer, 374–389.

[11] Jeff Howe. 2006. The rise of crowdsourcing. *Wired magazine* 14, 6 (2006), 1–4.

[12] Jon Kleinberg and Sigal Oren. 2011. Mechanisms for (mis) allocating scientific credit. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*. 529–538.

[13] Jon Kleinberg and Sigal Oren. 2022. Mechanisms for (Mis) allocating Scientific Credit. *Algorithmica* 84, 2 (2022), 344–378.

[14] Winter Mason and Duncan J Watts. 2009. Financial incentives and the" performance of crowds". In *Proceedings of the ACM SIGKDD workshop on human computation*. 77–85.

[15] Jessica L. Newman, Enrico Gerding, Enrico Marchioni, and Baharak Rastegari. 2025. Resource Task Games. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems* (Detroit, MI, USA) *(AAMAS '25)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1529–1537.

[16] Michael Wooldridge and Paul E Dunne. 2006. On the computational complexity of coalitional resource games. *Artificial Intelligence* 170, 10 (2006), 835–871.