Student Supervisor Matching : A New Approach

Abstract

Fair and efficient resource allocation is a fundamental challenge in multi-agent systems with applications in spectrum sharing, task assignment, server allocation, and IoT communication. We propose an iterative best response algorithm for reward allocation in congestion games, optimizing a global potential function to drive agents toward a local equilibrium. Each agent's reward is determined by its fixed value and a weight derived from its ranking in the reward's preference list. Agents iteratively compute potential gains from switching assignments, dynamically reconfiguring allocations to maximize system-wide utility. Our approach ensures convergence to a stable, envy-minimizing equilibrium while maintaining computational efficiency, making it well-suited for complex, real-world allocation problems.

1 Introduction

Efficient allocation of limited resources among competing agents is a fundamental problem in multi-agent systems, spanning domains such as wireless communication, computational resource management, task assignment, and IoT networking. In scenarios where multiple agents compete for rewards, fairness and efficiency become critical, necessitating a structured mechanism that balances preference satisfaction and optimal resource distribution. We present a formal allocation model that seeks to minimize total envy among agents while ensuring a proportional distribution of shared rewards based on predefined preference rankings. Each agent maintains a strict preference order over a set of rewards, influencing the utility derived from the allocation. The challenge lies in designing an allocation rule that guarantees fairness while adapting dynamically to real-world constraints where resource contention is inevitable.

Our model is particularly applicable in dynamic and competitive environments where prioritization is based on historical performance, skill levels, or bidding mechanisms. In wireless spectrum access, for example, secondary users opportunistically access licensed channels while avoiding interference with primary users, with prioritization influenced by reliability metrics. Similarly, in labor markets, workers compete for high-paying tasks, and employers prefer workers with specialized skills, leading to an allocation mechanism where tasks are assigned based on mutual ranking preferences. In cloud computing and distributed systems, jobs must bid for limited computational resources such as GPUs and CPUs, with high-priority jobs receiving greater compute power while lower-ranked jobs experience resource constraints. A similar allocation challenge arises in IoT networks, where devices transmitting data over shared wireless channels must be prioritized based on signal strength and reliability, ensuring minimal interference and efficient data transmission.

To address these challenges, we propose a structured allocation framework that formalizes reward distribution through a well-defined preference-based utility function. The model enforces constraints ensuring that each agent receives exactly one reward while allowing multiple agents to share a reward proportionally based on their preference weights. Envy, a key fairness metric, is explicitly quantified and minimized, preventing significant disparities in utility across agents. We formulate the problem as an optimization task, aiming to achieve a balance between fairness and efficiency in resource distribution. Our approach not only ensures equitable allocation but also aligns incentives across diverse competitive environments where agents learn and adapt over time based on historical interactions.

By providing a structured mathematical foundation for envy-minimizing allocation, our work contributes to the broader study of fair resource distribution in multi-agent systems. The proposed framework generalizes across various domains where competition, prioritization, and dynamic adaptation play a crucial role, offering a scalable and theoretically grounded solution to real-world allocation problems.

2 Related Work

Resource allocation in multi-agent systems is a long-studied problem with applications ranging from wireless communications and computational resource management to distributed decision-making. Early research in this area focused on efficient allocation under centralized control, but recent efforts have increasingly emphasized fairness and decentralization [1, 2]. These studies typically address challenges such as envy minimization and proportional fairness, forming the theoretical backbone of many modern allocation schemes.

Congestion games have emerged as a natural model for capturing the competition among self-interested agents over shared resources. In these games, each agent's cost or utility depends on the number of agents utilizing the same resource. Significant contributions have been made in analyzing best response dynamics in congestion games, including convergence properties and the efficiency of equilibria [3, 4]. In particular, the iterative best response framework has been shown to drive the system toward a local equilibrium, albeit with the challenge of balancing fairness against efficiency [5].

Fairness metrics in resource allocation have evolved to capture more nuanced objectives than pure efficiency. Notably, envy minimization—aiming to reduce the disparity in perceived fairness among agents—has been rigorously explored in various settings [1, 2, 6]. An alternative yet related measure is the maximum Nash welfare (MNW), which corresponds to the geometric mean of agents' utilities. MNW allocations have been proven to yield envy-freeness under certain conditions, particularly for divisible goods . Our work leverages a logarithmic potential function that is mathematically equivalent to maximizing the product of rewards, thereby linking it to Nash welfare optimization.

While our setting bears a superficial resemblance to the Fisher Market Equilibrium (FME) in that agents are assigned valuable resources (supervisors) based on preferences, it is fundamentally different in both structure and objective. Unlike FME, agents in our model do not possess budgets, and there is no notion of prices or market-clearing. Allocations are indivisible and binary—each agent is assigned exactly one supervisor—making the problem inherently discrete and non-convex, in contrast to the continuous and convex nature of FME. Moreover, the utility functions in our model are non-concave and interdependent, as an agent's utility depends not only on their assignment but also on how the supervisor's value is split among others, violating FME's separability and monotonicity assumptions. Finally, while FME seeks to maximize utilities under budget constraints, our framework aims to minimize total envy among agents, emphasizing fairness in a way that diverges significantly from the equilibrium-based approach of Fisher markets.

In the realm of congestion games, fair intervention strategies have also been proposed. For instance, researchers have developed mechanisms to improve equilibrium quality through fair taxation or incentive schemes, ensuring that interventions remain computationally efficient while enhancing fairness [7, 8]. Such approaches often introduce artificial currency or weight-dependent pricing policies that balance individual incentives with collective outcomes.

Finally, iterative optimization and learning-based approaches have recently been applied to resource allocation in dynamic environments. These methods enable agents to adapt to changing conditions by balancing exploration and exploitation, a key requirement in multi-agent multi-armed bandit settings [9, 10, 11, 12, 13]. By continuously updating their strategies based on observed rewards, agents can implicitly converge to stable matchings that reflect both their preferences and those of the resources, paving the way for decentralized coordination in complex systems.

Overall, the literature highlights a rich interplay between fairness, efficiency, and stability in multi-agent resource allocation. Our work builds on these foundations by employing an iterated best response mechanism guided by a global potential function, and it sets the stage for future extensions into more dynamic, learning-based environments.

3 Model Formulation

Given a set of supervisors $S = \{s_1, s_2, \dots, s_k\}$ with fixed values $v_i > 0$ for each supervisor i and a strict preference ordering over a set of agents (students) $A = \{a_1, a_2, \dots, a_n\}$, where $n >> k$, we aim to allocate supervisors among agents such that the total envy among the agents is minimized.

3.1 Supervisor Preferences and Weights

Each supervisor s_i has a preference ordering over the agents and is denoted by:

$$a^i_{p_1} \succ a^i_{p_2} \succ \cdots \succ a^i_{p_n},$$

where $a^i_{p_1}$ is the most preferred and $a^i_{p_n}$ is the least preferred agent for supervisor i. The weight of supervisor s_i for an agent a_j is defined as:

$$w_{i,j} = \gamma^{\mathrm{rank}_i(a_j)}, \quad \gamma < 1. \tag{1}$$

where $\text{rank}_i(a_j)$ denotes the position of a_j in the preference list of s_i.

3.2 Allocation Rule

Each agent must be assigned exactly one supervisor, and each supervisor must be assigned to at least one agent. This assumption of each supervisor being allotted atleast one agent might not minimize envy in general. But later we prove that under some mild assumption, a minimizing envy solution will always satisfy this condition. If multiple agents are assigned the same supervisor s_i, the reward of the supervisor s_i denoted by v_i is divided proportionally to their weights. Let S_i denote the set of agents assigned to supervisor s_i:

$$S_i = \{a_j \in A \mid x_{i,j} = 1\},$$

where $x_{i,j}$ is a binary decision variable:

$$x_{i,j} = \begin{cases} 1, & \text{if agent } a_j \text{ is assigned supervisor } s_i, \\ 0, & \text{otherwise.} \end{cases}$$

The utility received by agent a_j when assigned supervisor s_i is:

$$U_{i,j} = \frac{w_{i,j} v_i}{\sum\limits_{a_m \in S_i} w_{i,m}}.$$

Since each agent receives exactly one supervisor, the overall utility of agent a_j is:

$$U_j(x) = \sum_{i=1}^{k} x_{i,j} U_{i,j}.$$

Here, x is an allocation vector.

3.3 Envy Definition

For any two agents a_j and a_m, the envy of agent a_j toward agent a_m is defined as:

$$E_{j,m}(x) = \max\{0, U_m(x) - U_j(x)\}.$$

3.4 Objective

The primary objective of our allocation mechanism is to find an assignment of supervisors to agents that minimizes the maximum envy between any two agents. Formally, we aim to solve:

$$\min_x \max_{j,m} E_{j,m}(x),$$

where the envy between agents j and m is defined as

$$E_{j,m}(x) = \max\{0, U_m - U_j\}.$$

This formulation ensures that the worst-case envy across all agent pairs is as small as possible, leading to a fairer allocation.

3.5 Optimization Problem

The objective is to determine an assignment $\{x_{i,j}\}$ that minimizes the maximum envy between any pair of agents. We formulate this as an Integer Linear Programming (ILP) problem.

3.5.1 Decision Variables

- $x_{i,j} \in \{0, 1\}$: Binary variable indicating whether agent a_j is assigned to supervisor s_i.
- $z \geq 0$: Variable representing the maximum envy between any pair of agents.
- $E_{j,m} \geq 0$: Variable representing the envy of agent a_j toward agent a_m.
- $U_j \geq 0$: Variable representing the utility of agent a_j.
- $y_i \geq 0$: Variable representing the total weighted sum for supervisor s_i.
- $\delta_{i,j} \in \{0, 1\}$: Binary variable indicating whether agent a_j receives utility from supervisor s_i.
- $\hat{U}_{i,j} \geq 0$: Variable representing the utility contribution of supervisor s_i to agent a_j.

3.5.2 Objective Function

We aim to minimize the maximum envy between any pair of agents:

$$\min z \tag{2}$$

3.5.3 Constraints

1. Each agent must be assigned exactly one supervisor:

$$\sum_{i=1}^{k} x_{i,j} = 1, \quad \forall j \in \{1, 2, \ldots, n\} \tag{3}$$

2. Each supervisor must be assigned to at least one agent:

$$\sum_{j=1}^{n} x_{i,j} \geq 1, \quad \forall i \in \{1, 2, \ldots, k\} \tag{4}$$

3. Calculate the total weight assigned to each supervisor:

$$y_i = \sum_{j=1}^{n} w_{i,j} \cdot x_{i,j}, \quad \forall i \in \{1, 2, \ldots, k\} \tag{5}$$

4. Prevent division by zero when calculating utilities:

$$y_i \geq \epsilon \cdot \sum_{j=1}^{n} x_{i,j}, \quad \forall i \in \{1, 2, \ldots, k\} \tag{6}$$

where ϵ is a small positive constant.

5. Utility calculation constraints:

$$\delta_{i,j} \leq x_{i,j}, \quad \forall i \in \{1, 2, \ldots, k\}, \forall j \in \{1, 2, \ldots, n\} \tag{7}$$

$$\hat{U}_{i,j} \leq M \cdot \delta_{i,j}, \quad \forall i \in \{1, 2, \ldots, k\}, \forall j \in \{1, 2, \ldots, n\} \tag{8}$$

where M is a large positive constant.

6. Define the relationship between utility contribution and assignment:

$$\hat{U}_{i,j} \cdot y_i \leq w_{i,j} \cdot v_i \cdot x_{i,j}, \quad \forall i \in \{1, 2, \ldots, k\}, \forall j \in \{1, 2, \ldots, n\} \tag{9}$$

$$\hat{U}_{i,j} \cdot y_i \geq w_{i,j} \cdot v_i \cdot x_{i,j} - M \cdot (1 - x_{i,j}), \quad \forall i \in \{1, 2, \ldots, k\}, \forall j \in \{1, 2, \ldots, n\} \tag{10}$$

These constraints ensure that when $x_{i,j} = 1$, $\hat{U}_{i,j} = \frac{w_{i,j} \cdot v_i}{y_i}$.

7. Total utility of each agent:

$$U_j = \sum_{i=1}^{k} \hat{U}_{i,j}, \quad \forall j \in \{1, 2, \ldots, n\} \tag{11}$$

8. Define envy between agents:

$$E_{j,m} \geq U_m - U_j, \quad \forall j, m \in \{1, 2, \ldots, n\}, j \neq m \tag{12}$$

$$E_{j,m} \geq 0, \quad \forall j, m \in \{1, 2, \ldots, n\}, j \neq m \tag{13}$$

9. Define maximum envy:

$$z \geq E_{j,m}, \quad \forall j, m \in \{1, 2, \ldots, n\}, j \neq m \tag{14}$$

3.5.4 Linearization of Bilinear Constraints

The constraints (7) and (8) involve bilinear terms $\hat{U}_{i,j} \cdot y_i$, which are not linear. To linearize these constraints, we can use the following approach:

1. Replace $\hat{U}_{i,j} \cdot y_i$ with a new variable $V_{i,j}$.

2. Add McCormick envelope constraints to ensure $V_{i,j} = \hat{U}_{i,j} \cdot y_i$.

Let $\underline{y_i}$ and $\overline{y_i}$ be lower and upper bounds on y_i, and $\underline{\hat{U}_{i,j}}$ and $\overline{\hat{U}_{i,j}}$ be lower and upper bounds on $\hat{U}_{i,j}$. The McCormick envelope constraints are:

$$V_{i,j} \geq \underline{y_i} \cdot \hat{U}_{i,j} + y_i \cdot \underline{\hat{U}_{i,j}} - \underline{y_i} \cdot \underline{\hat{U}_{i,j}} \tag{15}$$

$$V_{i,j} \geq \overline{y_i} \cdot \hat{U}_{i,j} + y_i \cdot \overline{\hat{U}_{i,j}} - \overline{y_i} \cdot \overline{\hat{U}_{i,j}} \tag{16}$$

$$V_{i,j} \leq \underline{y_i} \cdot \hat{U}_{i,j} + y_i \cdot \overline{\hat{U}_{i,j}} - \underline{y_i} \cdot \overline{\hat{U}_{i,j}} \tag{17}$$

$$V_{i,j} \leq \overline{y_i} \cdot \hat{U}_{i,j} + y_i \cdot \underline{\hat{U}_{i,j}} - \overline{y_i} \cdot \underline{\hat{U}_{i,j}} \tag{18}$$

Now, constraints (7) and (8) become:

$$V_{i,j} \leq w_{i,j} \cdot v_i \cdot x_{i,j}, \quad \forall i \in \{1, 2, \ldots, k\}, \forall j \in \{1, 2, \ldots, n\} \tag{19}$$

$$V_{i,j} \geq w_{i,j} \cdot v_i \cdot x_{i,j} - M \cdot (1 - x_{i,j}), \quad \forall i \in \{1, 2, \ldots, k\}, \forall j \in \{1, 2, \ldots, n\} \tag{20}$$

3.6 Social Welfare

Social welfare is defined as the sum of the utilities of all the students i.e.

$$SW(x) = \sum_j U_j(x)$$

Social welfare of an allocation x is defined as the sum of the utilities of all students:

$$SW(x) = \sum_{j=1}^{n} U_j(x) = \sum_{j=1}^{n} \sum_{i=1}^{k} x_{i,j} U_{i,j}. \tag{21}$$

Substituting the formula for $U_{i,j}$:

$$SW(x) = \sum_{j=1}^{n} \sum_{i=1}^{k} x_{i,j} \frac{w_{i,j} v_i}{\sum_{a_m \in S_i} w_{i,m}}. \tag{22}$$

Rearranging the summation by grouping terms for each supervisor s_i:

$$SW(x) = \sum_{i=1}^{k} \sum_{j \in S_i} \frac{w_{i,j} v_i}{\sum_{a_m \in S_i} w_{i,m}}. \tag{23}$$

Since every agent contributes exactly once to the sum under their assigned supervisor, and the fraction sums to 1 for each supervisor:

$$\sum_{j \in S_i} \frac{w_{i,j}}{\sum_{a_m \in S_i} w_{i,m}} = 1, \tag{24}$$

we obtain:

$$SW(x) = \sum_{i=1}^{k} v_i. \tag{25}$$

Since SW only depends on the fixed supervisor values v_i and not on the specific allocation of agents (as long as each supervisor has at least one agent), social welfare remains constant across all allocations.

Thus, it can be seen that social welfare is same for all the allocation if our assumption that all supervisor has been allocated atleast one agent.

Interestingly, since the total social welfare $SW(x) = \sum_i v_i$ remains constant across all feasible allocations (assuming each supervisor has at least one agent), our setting exhibits properties akin to a *constant-sum game*. In such games, the total utility in the system is fixed, implying that one agent's gain necessarily results in another's loss. This structure highlights that social welfare alone cannot differentiate between good and bad allocations, as it remains unchanged even in highly unfair or envy-inducing scenarios. Moreover, constant-sum games have Pareto optimal solutions, where no agent can be made better off without making another worse off. However, this limitation motivates the need for alternative analytical tools—such as potential functions—to study the system's equilibrium behavior and fairness properties.

3.7 Stability or Equilibria

The algorithm follows best-response dynamics, meaning each agent updates their allocation to increase the potential function $\Phi(U)$. This follows a best-response dynamic, where each agent improves the global potential function at every step.

Since the potential function $\Phi(U)$ strictly increases during each best-response update, the process is guaranteed to move towards a local maximum of $\Phi(U)$.

The potential function $\Phi(U)$ is bounded from above because:

$$\Phi(U) = \sum_{i=1}^{n} \log(U_i) \tag{26}$$

is an increasing function , and the utility values U_i are finite and bounded (since rewards have fixed values). Since $\Phi(U)$ strictly increases at each step and is bounded, the process must eventually reach a fixed point, meaning that no agent can further increase the function.

Once the algorithm converges, no further updates occur. This means that for every agent a_i, switching to another reward will not increase the potential function .

In other words, each agent is playing a best response to the current allocation, implying that no unilateral deviation is beneficial. This satisfies the condition for a equilibrium:

$$U_i(s_i, s_{-i}) \geq U_i(s'_i, s_{-i}) \quad \forall i, \forall s'_i \tag{27}$$

where s_i is the current reward allocation and s'_i is an alternative allocation.

4 Proposed Algorithm

We next show that the proposed potential function based approach leads to a equilibria solution which also results in minimum envy.

For this we propose an iterative best response algorithm for supervisor allocation in congestion games. The underlying idea is to let agents unilaterally update their supervisor assignments in order to maximize a global potential function. The potential function is defined as:

$$\Phi(\mathbf{U}) = \sum_{i=1}^{n} \log(U_i),$$

where U_i denotes the utility received by agent a_i. Here, each agent's utility U_i is computed based on the supervisor's fixed value and the weights derived from the agent's rank in the reward's preference list. Recall that the weight for agent a_j for supervisor s_i is given by:

$$w_{i,j} = \gamma^{\text{rank}_i(a_j)}, \quad \gamma < 1.$$

At each iteration, every agent evaluates the potential gain from switching to any candidate reward. An agent a_i computes a temporary potential function Φ_{temp} that would result if they switched their assignment, and then chooses the reward that maximizes the overall potential. This process is repeated until no agent can improve the potential by switching supervisors, implying convergence to a local equilibrium.

4.1 Algorithm Pseudocode

Algorithm 1 Fair Reward Assignment via Iterated Best Response

procedure ASSIGNREWARDS
    Input: n agents, k supervisors
    Initialize assignments for n agents randomly with supervisors from 1 to k such as each agent is allocated exactly one supervisor
    Generate rank array P of n elements that contains a random permutation of agents a_1 to a_n which serves as the order in which agents go on improving global potential function.
    repeat
        for each agent $i = 1, \ldots, n$ **do**
            $\Phi^* \leftarrow -\infty$
            $j^* \leftarrow$ current supervisor of agent i
            for each candidate supervisor $j = 1, \ldots, k$ **do**
                Compute temporary potential function Φ_{temp} if agent i switches to supervisor j
                if $\Phi_{\text{temp}} > \Phi^*$ **then**
                    $\Phi^* \leftarrow \Phi_{\text{temp}}$
                    $j^* \leftarrow j$
                end if
            end for
            Update student i's assignment to supervisor j^*
        end for
    until no assignments change
end procedure

The algorithm terminates when no student can further increase the potential function by changing their reward. This iterative process guarantees convergence to a (local) equilibrium for the reward allocation game.

5 Related Proofs

5.1 Maximization of $\sum \ln(U_j)$ Ensures Minimization of Envy

Theorem 1. *Maximizing the function*

$$F = \sum_{j=1}^{n} \ln(U_j)$$

ensures that the envy

$$E_{j,m} = \max\{0, U_m - U_j\}.$$

is minimized.

Proof. We show that maximizing F leads to an equal allocation of utilities, which in turn minimizes envy. By the Arithmetic Mean-Geometric Mean (AM-GM) inequality, for positive numbers $U_1, U_2, \ldots, U_n$ we have

$$\prod_{j=1}^{n} U_j \leq \left(\frac{1}{n} \sum_{j=1}^{n} U_j \right)^n,$$

with equality if and only if $U_1 = U_2 = \cdots = U_n$. Taking the natural logarithm of both sides yields

$$\sum_{j=1}^{n} \ln(U_j) \leq n \ln \left(\frac{1}{n} \sum_{j=1}^{n} U_j \right),$$

with equality if and only if all U_j are equal. Thus, under any fixed total (or average) utility constraint, F is maximized when

$$U_1 = U_2 = \cdots = U_n = c,$$

for some constant c. In this scenario, for any pair (j, m) we have

$$\max(0, U_m - U_j) = \max(0, c - c) = 0.$$

Therefore, the envy is

$$E_{\mathrm{j,m}} = 0,$$

which is the minimum possible value. Hence, maximizing $\sum_{j=1}^{n} \ln(U_j)$ leads to the minimization of envy.

However, in practice, perfect equality in utility values (i.e., $U_1 = U_2 = \cdots = U_n$) may not be attainable due to the discrete nature of the supervisor allocation problem and the constraints imposed by the preference structures. Even in such cases, our algorithm's objective function $F = \sum_{j=1}^{n} \ln(U_j)$ still provides strong guarantees for minimizing envy.

When perfect equality is not attainable, the logarithmic function's strict concavity ensures that maximizing F inherently penalizes large disparities between utility values. Consider the first-order conditions for maximizing F with respect to the utilities U_j:

$$\frac{\partial F}{\partial U_j} = \frac{1}{U_j}$$

The larger the value of $\frac{1}{U_j}$, the more sensitive F is to changes in U_j. This means that agents with lower utilities have a proportionally larger impact on the objective function. Consequently, the algorithm will prioritize increasing the utilities of agents with the lowest values, leading to a more balanced distribution.

Furthermore, the logarithmic function has the property that

$$\lim_{U_j \to 0} \ln(U_j) = -\infty$$

This means that as any utility approaches zero, the objective function approaches negative infinity, creating a strong incentive to maintain positive utilities for all agents.

Given these properties, when the algorithm maximizes F, it implicitly minimizes the variance of the utility distribution subject to the constraints of the problem. This is because:

- The algorithm will try to equalize utilities as much as possible due to the concavity of the logarithm.
- The algorithm will strongly avoid allocations where some agents receive very low utilities.
- When perfect equality is not possible, the algorithm will find the allocation that minimizes the weighted sum of utility differences.

Moreover, maximizing $F = \sum_{j=1}^{n} \ln(U_j)$ is equivalent to maximizing the Nash social welfare function $NSW = \prod_{j=1}^{n} U_j$, as $\ln(NSW) = \sum_{j=1}^{n} \ln(U_j)$. The Nash social welfare function has been shown to balance efficiency and fairness concerns, and in our context, it helps ensure that the allocation minimizes envy even when perfect equality is not achievable.

In conclusion, while the theoretical minimum envy of zero may not be attainable in all instances, maximizing $F = \sum_{j=1}^{n} \ln(U_j)$ guarantees that the resulting allocation produces utility values that are as close to equal as possible, thereby minimizing total envy given the constraints of the problem. □

5.2 Proof of Algorithm Termination

Proof. We prove that the fair reward assignment algorithm terminates by showing that the potential function

$$\Phi(r) = \sum_{i=1}^{n} \ln(U_i)$$

(where U_i is the utility of agent a_i) strictly increases with each agent's best response update and that there exists an upper bound on this function.

Let $r^t = (r_1^t, r_2^t, \ldots, r_n^t)$ denote the reward allocation after iteration t, and let agent a_i be the one updating its strategy at iteration $t+1$. Suppose agent a_i switches from reward r_j to reward r_k at this step. The algorithm ensures that such a switch occurs only if it increases the agent's utility U_i.

Let S_j^t and S_k^t be the sets of agents assigned to rewards r_j and r_k, respectively, at iteration t. After agent a_i moves, these sets update to:

$$S_j^{t+1} = S_j^t \setminus \{a_i\}, \quad S_k^{t+1} = S_k^t \cup \{a_i\}.$$

The change in the potential function is given by

$$\begin{aligned}\Delta\Phi &= \Phi(r^{t+1}) - \Phi(r^t) \\ &= \left[\sum_{a \in S_j^{t+1}} \ln(U_a) - \sum_{a \in S_j^t} \ln(U_a)\right] + \left[\sum_{a \in S_k^{t+1}} \ln(U_a) - \sum_{a \in S_k^t} \ln(U_a)\right]. \end{aligned} \tag{28}$$

Since agent a_i performs a best response update, the increase in its utility guarantees that $\ln(U_i)$ increases; hence, $\Delta\Phi > 0$ unless a_i already has a best response (in which case $\Delta\Phi = 0$). The algorithm terminates when no agent can further improve its utility, i.e., when $\Delta\Phi = 0$ for all agents, corresponding to a stable allocation.

Moreover, because the number of possible reward assignments is finite (specifically, k^n if there are k rewards), and the potential function is bounded above (since the utilities U_i are bounded), the algorithm must terminate in a finite number of steps. □

5.3 Proof of Ensuring No Supervisors Unassigned

Proof. We ensure that every reward receives at least one agent by setting the reward values within the range:

$$\left(\frac{\gamma^2}{\gamma+\gamma^2} R_{\max}, R_{\max}\right]$$

where $0 < \gamma < 1$. By explicitly setting all rewards within this range, we ensure:

1. No reward is too small: The lower bound $\frac{\gamma^2}{\gamma+\gamma^2} R_{\max}$ is strictly positive, preventing rewards from being ignored.

2. No reward is too large: The upper bound $R_{\max}$ ensures that rewards remain balanced and prevent excessive clustering.

To prove that no reward remains unassigned at equilibrium, let us consider the contrapositive. Suppose at equilibrium, there exists a reward r_j that remains unassigned. Let $r_{\min}$ be the smallest reward value, which by our construction is:

$$r_{\min} > \frac{\gamma^2}{\gamma+\gamma^2} R_{\max}$$

Now, consider an agent a_i who is currently assigned to some reward r_k along with at least one other agent. In the best case scenario for a_i, it would be the second highest ranked agent for r_k (in a collision with the highest ranked agent). When sharing reward r_k of maximum value $R_{\max}$ with the highest ranked agent, the second highest agent would receive utility:

$$U_{k,i} = \frac{\gamma^2 \cdot R_{\max}}{\gamma + \gamma^2} = \frac{\gamma^2}{\gamma + \gamma^2} R_{\max}$$

If this agent were to switch to the unassigned reward r_j, its utility would be:

$$U_{j,i} = r_j > r_{\min} > \frac{\gamma^2}{\gamma + \gamma^2} R_{\max} = U_{k,i}$$

Thus, $U_{j,i} > U_{k,i}$, demonstrating that the agent would have an incentive to switch to the unassigned reward. This contradicts our assumption of being at equilibrium. Even in the most favorable collision scenario, the second highest ranked agent always has the incentive to shift to an unassigned reward because the ranges are defined specifically to ensure this property. This guarantees that no reward will remain unallocated at equilibrium. □

5.4 Complexity Analysis

Theorem 2. *Algorithm 1 has a worst-case time complexity of $O(n^2 kT)$, where n is the number of agents, k is the number of rewards, and T is the number of iterations until convergence.*

Proof. In each iteration, the algorithm performs the following operations:

- For each of the n agents, the algorithm evaluates k possible reward reassignments.
- For each evaluation, the algorithm needs to compute the new reward distribution, which takes $O(n)$ time to account for all agents sharing the same reward.

Therefore, each iteration has a time complexity of $O(n^2 k)$. Let T be the number of iterations until convergence. The worst-case time complexity of the algorithm is then $O(n^2 kT)$.

To bound T, we note that there are at most k^n possible assignments. In the worst case, the algorithm visits a significant portion of these assignments before convergence. However, in practice, the number of iterations is typically much smaller due to the monotonically increasing nature of the potential function.

For space complexity, the algorithm requires:

- $O(nk)$ space for storing the weight matrix $w_{i,j}$
- $O(n)$ space for current reward assignments
- $O(k)$ space for tracking agents assigned to each reward

Therefore, the overall space complexity is $O(nk)$. □

5.5 Bound On Envy

At equilibrium, we can establish a bound on the maximum envy that any agent may experience. This bound provides theoretical guarantees on the fairness properties of our allocation algorithm.

Theorem 3. *Let $E_{j,m}$ be the envy of agent a_j toward agent a_m at a equilibrium obtained by Algorithm 1, and let E_{OPT} be the minimum possible total envy. Then,*

$$E_{j,m} \leq U_j^{PE} \cdot \left(3 \cdot \frac{1}{\gamma^{n-1}} \cdot \frac{v_{max}}{v_{min}} - 1\right)$$

where v_{max} and v_{min} are the maximum and minimum supervisor values, respectively.

Proof. We begin with the potential function that drives our algorithm:

$$\Phi(U) = \sum_{i=1}^{n} \log(U_i)$$

where U_i is the utility received by agent a_i. This function is related to the Nash product, which has been shown to provide good fairness guarantees [14].

For any agent a_j, the utility received is:

$$U_j = \sum_{i=1}^{k} x_{i,j} \frac{w_{i,j} v_i}{\sum_{a_m \in S_i} w_{i,m}}$$

where $x_{i,j}$ equals 1 if agent a_j is assigned supervisor s_i and 0 otherwise, $w_{i,j}$ is the weight of agent a_j for supervisor s_i, v_i is the value of supervisor s_i, and S_i is the set of agents assigned to supervisor s_i.

Let r^* be the globally optimal supervisor allocation that minimizes total envy, and let r^{PE} be the equilibrium allocation obtained by our algorithm.

At equilibrium, no agent can unilaterally improve their utility by changing their strategy. For any agent a_j currently assigned to supervisor s_i, switching to any other supervisor $s_{i'}$ would not increase their utility.

Consider agent a_j and agent a_m where a_m is assigned to supervisor s_{i_m}. At equilibrium, agent a_j cannot improve their utility by switching to supervisor s_{i_m}. This implies:

$$U_j^{PE} \geq \frac{w_{i_m,j} v_{i_m}}{\sum_{a_l \in S_{i_m}} w_{i_m,l} + w_{i_m,j}}$$

Taking logarithms of both sides:

$$\log(U_j^{PE}) \geq \log\left(\frac{w_{i_m,j} v_{i_m}}{\sum_{a_l \in S_{i_m}} w_{i_m,l} + w_{i_m,j}}\right)$$

We can rewrite this as:

$$\log(U_j^{PE}) \geq \log(w_{i_m,j}) + \log(v_{i_m}) - \log\left(\sum_{a_l \in S_{i_m}} w_{i_m,l} + w_{i_m,j}\right)$$

Now, let's consider agent a_m's utility at equilibrium:

$$U_m^{PE} = \frac{w_{i_m,m} v_{i_m}}{\sum_{a_l \in S_{i_m}} w_{i_m,l}}$$

Taking logarithms:

$$\log(U_m^{PE}) = \log(w_{i_m,m}) + \log(v_{i_m}) - \log\left(\sum_{a_l \in S_{i_m}} w_{i_m,l}\right)$$

Subtracting the first inequality from this equation:

$$\begin{aligned}
\log(U_m^{PE}) - \log(U_j^{PE}) &\leq \log(w_{i_m,m}) - \log(w_{i_m,j}) \\
&\quad + \log\left(\sum_{a_l \in S_{i_m}} w_{i_m,l} + w_{i_m,j}\right) - \log\left(\sum_{a_l \in S_{i_m}} w_{i_m,l}\right) \\
&= \log\left(\frac{w_{i_m,m}}{w_{i_m,j}}\right) + \log\left(\frac{\sum_{a_l \in S_{i_m}} w_{i_m,l} + w_{i_m,j}}{\sum_{a_l \in S_{i_m}} w_{i_m,l}}\right) \\
&= \log\left(\frac{w_{i_m,m}}{w_{i_m,j}}\right) + \log\left(1 + \frac{w_{i_m,j}}{\sum_{a_l \in S_{i_m}} w_{i_m,l}}\right)
\end{aligned}$$

Using the property $\log(1+x) \leq x$ for $x > -1$:

$$\log(U_m^{PE}) - \log(U_j^{PE}) \leq \log\left(\frac{w_{i_m,m}}{w_{i_m,j}}\right) + \frac{w_{i_m,j}}{\sum_{a_l \in S_{i_m}} w_{i_m,l}}$$

$$\leq \log\left(\frac{w_{i_m,m}}{w_{i_m,j}}\right) + 1$$

since $\frac{w_{i_m,j}}{\sum_{a_l \in S_{i_m}} w_{i_m,l}} \leq 1$.

Recalling that $w_{i,j} = \gamma^{\text{rank}_i(a_j)}$ with $\gamma < 1$, the ratio $\frac{w_{i_m,m}}{w_{i_m,j}}$ can be expressed as:

$$\frac{w_{i_m,m}}{w_{i_m,j}} = \frac{\gamma^{\text{rank}_{i_m}(a_m)}}{\gamma^{\text{rank}_{i_m}(a_j)}} = \gamma^{\text{rank}_{i_m}(a_m) - \text{rank}_{i_m}(a_j)}$$

In the worst case, agent a_m is ranked first ($\text{rank}_{i_m}(a_m) = 1$) and agent a_j is ranked last ($\text{rank}_{i_m}(a_j) = n$) for supervisor s_{i_m}. This gives:

$$\frac{w_{i_m,m}}{w_{i_m,j}} \leq \frac{\gamma^1}{\gamma^n} = \gamma^{1-n} = \frac{1}{\gamma^{n-1}}$$

Therefore:

$$\log(U_m^{PE}) - \log(U_j^{PE}) \leq \log\left(\frac{1}{\gamma^{n-1}}\right) + 1 = (n-1)\log\left(\frac{1}{\gamma}\right) + 1$$

Now, we relate this to the ratio of utilities:

$$\log\left(\frac{U_m^{PE}}{U_j^{PE}}\right) \leq (n-1)\log\left(\frac{1}{\gamma}\right) + 1$$

Taking exponentials of both sides:

$$\frac{U_m^{PE}}{U_j^{PE}} \leq e^{(n-1)\log\left(\frac{1}{\gamma}\right)+1} = e \cdot \left(\frac{1}{\gamma}\right)^{n-1}$$

$$\leq 3 \cdot \frac{1}{\gamma^{n-1}}$$

where we use the fact that $e < 3$.

We also account for different supervisor values. In the worst case, agent a_m is assigned to the highest-value supervisor ($v_{\max}$) and agent a_j is assigned to the lowest-value supervisor ($v_{\min}$). This gives:

$$\frac{U_m^{PE}}{U_j^{PE}} \leq 3 \cdot \frac{1}{\gamma^{n-1}} \cdot \frac{v_{\max}}{v_{\min}}$$

Finally, we relate this to the envy. The envy of agent a_j toward agent a_m is:

$$E_{j,m} = \max\{0, U_m^{PE} - U_j^{PE}\} = U_j^{PE} \cdot \left(\frac{U_m^{PE}}{U_j^{PE}} - 1\right)$$

$$\leq U_j^{PE} \cdot \max\left\{0, \frac{U_m^{PE}}{U_j^{PE}} - 1\right\} \leq U_j^{PE} \cdot \left(3 \cdot \frac{1}{\gamma^{n-1}} \cdot \frac{v_{\max}}{v_{\min}} - 1\right)$$

Since $\gamma < 1$ and we typically have $n \gg k$ (many more agents than supervisors), we can expect $\frac{1}{\gamma^{n-1}}$ to be large. Therefore, the term inside the parentheses will be positive, and we obtain:

$$E_{j,m} \leq U_j^{PE} \cdot \left(3 \cdot \frac{1}{\gamma^{n-1}} \cdot \frac{v_{\max}}{v_{\min}} - 1\right)$$

This completes the proof. □

Note 1 (Generalizing the weight-ratio bound). Define for each supervisor s_i the maximum and minimum weights she may assign:

$$W_i^{\max} = \max_{a_j} w_{i,j}, \qquad W_i^{\min} = \min_{a_j} w_{i,j}.$$

Further let

$$W^{\max} = \max_{1\leq i\leq k} W_i^{\max}, \qquad W^{\min} = \min_{1\leq i\leq k} W_i^{\min}.$$

Then in the same style as above, the term $\gamma^{1-n} = 1/\gamma^{n-1}$ can be replaced by the more general ratio

$$\frac{W^{\max}}{W^{\min}},$$

giving the following *general bound* on envy:

$$E_{j,m} \;\le\; U_j^{PE} \;\cdot\; \left(3\,\frac{W^{\max}}{W^{\min}}\,\frac{v_{\max}}{v_{\min}} \;-\; 1\right).$$

Moreover, if we restrict attention only to the weights actually used in the equilibrium assignment, define

$$\widehat{W}_i^{\max} \;=\; \max_{a_j\colon x_{i,j}=1} w_{i,j}, \quad \widehat{W}_i^{\min} \;=\; \min_{a_j\colon x_{i,j}=1} w_{i,j},$$

and set

$$\widehat{W}^{\max} = \max_i \widehat{W}_i^{\max}, \quad \widehat{W}^{\min} = \min_i \widehat{W}_i^{\min}, \quad v_{\max}^{\text{assign}} = \max_i v_i, \quad v_{\min}^{\text{assign}} = \min_i v_i.$$

Then the envy can be bounded more tightly by

$$E_{j,m} \;\le\; U_j^{PE} \;\cdot\; \left(3\,\frac{\widehat{W}^{\max}}{\widehat{W}^{\min}}\,\frac{v_{\max}^{\text{assign}}}{v_{\min}^{\text{assign}}} \;-\; 1\right).$$

This shows how the bound improves when only the *actually assigned* weights and supervisor values are taken into account.

5.6 Analysis of Maximum Utility Gain at Equilibrium

In this subsection, we further analyze the quality of the equilibrium obtained by our algorithm by establishing bounds on the maximum potential utility gain that agents could achieve from unilateral deviations.

Given the bound established in Theorem 3, we know that at equilibrium, the ratio between utilities of agents a_m and a_j is bounded by:

$$\frac{U_m^{PE}}{U_j^{PE}} \le 3 \cdot \frac{1}{\gamma^{n-1}} \cdot \frac{v_{max}}{v_{min}}$$

This inequality provides a lower bound on agent a_j's utility:

$$U_j^{PE} \ge \frac{U_m^{PE}}{3 \cdot \frac{1}{\gamma^{n-1}} \cdot \frac{v_{max}}{v_{min}}}$$

5.6.1 Utility Calculations at Equilibrium

At equilibrium, an agent a_j assigned to supervisor s_i receives utility:

$$U_j^{PE} = \frac{w_{i,j} \cdot v_i}{\sum_{a_l \in S_i} w_{i,l}}$$

Similarly, agent a_m assigned to supervisor s_k has utility:

$$U_m^{PE} = \frac{w_{k,m} \cdot v_k}{\sum_{a_l \in S_k} w_{k,l}}$$

5.6.2 Maximum Potential Utility Gain

To analyze the maximum potential utility gain, we consider a scenario where agent a_j shifts from supervisor s_i to supervisor s_k (the supervisor of agent a_m). After shifting, agent a_j's utility would be:

$$U_j^{SHIFT} = \frac{w_{k,j} \cdot v_k}{\sum_{a_l \in S_k} w_{k,l} + w_{k,j}}$$

The maximum utility gain agent a_j can achieve is:

$$U_{max} = U_j^{SHIFT} - \min(U_j^{PE})$$

Using our lower bound for U_j^{PE}, we can derive:

$$U_{max} = \frac{w_{k,j} \cdot v_k}{\sum_{a_l \in S_k} w_{k,l} + w_{k,j}} - \frac{U_m^{PE}}{3 \cdot \frac{1}{\gamma^{n-1}} \cdot \frac{v_{max}}{v_{min}}}$$

Substituting the expression for U_m^{PE}:

$$U_{max} = \frac{w_{k,j} \cdot v_k}{\sum_{a_l \in S_k} w_{k,l} + w_{k,j}} - \frac{w_{k,m} \cdot v_k}{\sum_{a_l \in S_k} w_{k,l} \cdot 3 \cdot \frac{1}{\gamma^{n-1}} \cdot \frac{v_{max}}{v_{min}}}$$

For simplicity, let's define:

$$W_k = \sum_{a_l \in S_k} w_{k,l} \tag{29}$$

$$\alpha = 3 \cdot \frac{1}{\gamma^{n-1}} \cdot \frac{v_{max}}{v_{min}} \tag{30}$$

Then:

$$U_{max} = v_k \left[\frac{w_{k,j}}{W_k + w_{k,j}} - \frac{w_{k,m}}{W_k \cdot \alpha} \right]$$

This formula reveals several insights:

1. The potential gain scales linearly with v_k, indicating that higher-value supervisors offer greater potential improvement.

2. The gain depends on the ratio of agent a_j's weight ($w_{k,j}$) versus agent a_m's weight ($w_{k,m}$) for supervisor s_k.

3. The denominator $W_k + w_{k,j}$ represents the "congestion" effect after shifting. As W_k increases, the potential gain decreases.

4. The bound coefficient α increases with larger n, smaller γ, and larger ratio $\frac{v_{max}}{v_{min}}$.

5. At a true equilibrium, U_{max} should be non-positive, confirming that no agent can unilaterally improve their utility.

This analysis helps quantify the quality of the equilibrium reached by our algorithm and identifies factors that affect the potential utility gain for agents considering shifts in their strategies.

5.7 Relationship between Nash Social Welfare and Envy

Theorem 4. *As the Nash social welfare increases, the maximum pairwise envy between agents monotonically decreases.*

Proof. Consider the Nash social welfare function defined as:

$$\mathrm{NSW}(x) = \prod_{j=1}^{n} U_j(x) \tag{31}$$

where U_j represents the utility of agent j, and n is the total number of agents.

Let NSW_1 and NSW_2 be two different allocations with $\mathrm{NSW}_1 < \mathrm{NSW}_2$. We will prove that the maximum envy in NSW_2 is less than or equal to the maximum envy in NSW_1.

First, recall the definition of envy between agents j and m:

$$E_{j,m} = \max\{0, U_m - U_j\} \tag{32}$$

By the Arithmetic Mean-Geometric Mean (AM-GM) inequality, we know that:

$$\frac{1}{n}\sum_{j=1}^{n} U_j \geq \left(\prod_{j=1}^{n} U_j\right)^{1/n} \tag{33}$$

with equality if and only if $U_1 = U_2 = \cdots = U_n$.

As $\mathrm{NSW}_2 > \mathrm{NSW}_1$, the AM-GM inequality suggests that the utility distribution in NSW_2 is more uniform compared to NSW_1.

Consider the logarithmic transformation of the Nash social welfare:

$$\log(\mathrm{NSW}) = \sum_{j=1}^{n} \log(U_j) \tag{34}$$

The concavity of the logarithmic function implies that a higher Nash social welfare corresponds to a more balanced utility distribution. This balance directly translates to reduced envy between agents.

Intuitively, as the Nash social welfare increases:

- The variance of utility values decreases
- The maximum disparity between agent utilities reduces
- The probability of large envy between any pair of agents diminishes

Hence, we can conclude that:

$$\mathrm{NSW}_2 > \mathrm{NSW}_1 \implies \max(E_{j,m} \text{ in } \mathrm{NSW}_2) \leq \max(E_{j,m} \text{ in } \mathrm{NSW}_1) \tag{35}$$

□

Sufficient Conditions for Increasing Nash Social Welfare

Several conditions can be identified that lead to increased Nash social welfare, thereby reducing envy:

1. **Balanced Allocation**: Allocations that distribute supervisors more evenly among agents with similar preference ranks tend to increase Nash social welfare. Mathematically, this corresponds to minimizing the variance of utility values:

$$\mathrm{Var}(U_1, U_2, \ldots, U_n) = \frac{1}{n}\sum_{j=1}^{n}(U_j - \bar{U})^2 \tag{36}$$

   where $\bar{U}$ is the mean utility across all agents.

2. **Preference-Aligned Matching**: When agents are matched with supervisors who rank them highly, the resulting $w_{i,j}$ values increase, leading to higher individual utilities and thus higher Nash social welfare.

3. **Optimal Congestion Control**: Limiting the number of agents assigned to high-value supervisors can prevent excessive dilution of utilities and maintain a higher Nash social welfare.

4. **Strategic Parameter Selection**: The choice of γ significantly affects the Nash social welfare and resulting envy, as discussed below.

The Role of γ in Nash Social Welfare and Envy

The parameter γ plays a crucial role in determining both Nash social welfare and envy levels:

- **Effect on Weight Distribution**: Recall that $w_{i,j} = \gamma^{\mathrm{rank}_i(a_j)}$ where $\gamma < 1$. As γ approaches 1, the weights become more evenly distributed across agents, regardless of their rank.

- **When** $\gamma \to 1$: The ratio between weights of differently ranked agents diminishes:

$$\lim_{\gamma \to 1} \frac{w_{i,j}}{w_{i,k}} = \lim_{\gamma \to 1} \gamma^{\mathrm{rank}_i(a_j) - \mathrm{rank}_i(a_k)} = 1 \tag{37}$$

  This creates a more egalitarian distribution of supervisor time, resulting in lower envy but potentially reduced efficiency if truly higher-ranked agents deserve more attention.

- **When** $\gamma \ll 1$: The weight distribution becomes highly skewed, with top-ranked agents receiving significantly higher weights:

$$\lim_{\gamma \to 0} \frac{w_{i,1}}{w_{i,2}} = \lim_{\gamma \to 0} \frac{\gamma^1}{\gamma^2} = \lim_{\gamma \to 0} \frac{1}{\gamma} = \infty \tag{38}$$

  This can lead to higher envy among lower-ranked agents but may maximize efficiency if rank accurately reflects productivity.

In student-supervisor matching specifically, γ can be interpreted as determining how supervisor time is distributed among students of different ranks. When γ is closer to 1, supervisor time is more evenly distributed, reducing envy but potentially not optimizing productivity. When γ is smaller, supervisor time is concentrated among higher-ranked students, potentially optimizing productivity but at the cost of higher envy among lower-ranked students.

This completes the proof that increasing Nash social welfare monotonically reduces the maximum envy between agents.

Remark 1. While the proof establishes a theoretical relationship, practical implementations may have constraints that prevent perfect envy minimization. The proposed algorithm provides an approximation that balances efficiency and fairness.

5.8 Global vs Local Optima in Reward Allocation

The iterative best-response algorithm proposed in this work converges to a local equilibrium, which may deviate from the globally optimal solution. Understanding the relationship between global and local optima is crucial for assessing the algorithm's performance and limitations.

5.8.1 Characteristics of Local Optima

Local optima emerge when no individual agent can unilaterally improve their utility by changing their current assignment. While these equilibria ensure a form of stability, they do not guarantee the globally most efficient or fair allocation. The potential function $\Phi(U) = \sum_{i=1}^{n} \log(U_i)$ drives agents toward local maxima, but these may represent only a subset of all possible allocation configurations.

5.8.2 Strategies for Approaching Global Optimality

Several approaches can be employed to mitigate the limitations of local optima:

1. **Randomized Initialization:** By initializing agent assignments randomly and running multiple trials, we increase the probability of discovering different local optima. Comparing outcomes across these trials can provide insights into the solution landscape.

2. **Simulated Annealing:** Introducing probabilistic mechanisms that occasionally accept utility-reducing moves can help escape local equilibria. This approach allows the algorithm to explore a broader solution space at the expense of computational complexity.

3. **Multi-Stage Optimization:** Implementing a hierarchical optimization process where initial coarse allocations are progressively refined can help navigate toward more globally efficient solutions.

5.8.3 Theoretical Limitations

The convergence to local optima is fundamentally tied to the algorithm's best-response dynamics. Subsection 5.2 guarantees convergence to a local equilibrium, but does not ensure global optimality. This inherent limitation stems from the non-convex nature of the reward allocation problem.

Mathematically, for an allocation configuration r^*, we can express the gap between local and global optima as:

$$\Delta_{opt} = \max_{r \in \mathcal{R}} SW(r) - SW(r^*) \tag{39}$$

where $SW(r)$ represents the social welfare for allocation r, and $\mathcal{R}$ denotes the set of all possible allocations.

5.8.4 Future Enhancements

Potential avenues for future research include:

- Developing meta-heuristics that can more effectively escape local optima
- Designing theoretical bounds on the suboptimality of local equilibria
- Exploring alternative potential functions that might provide better global convergence properties

By systematically investigating these strategies, researchers can develop more sophisticated allocation mechanisms that balance computational efficiency with global optimality.

6 Introduction of Supervisor Utility

To capture the preferences and load-balancing considerations of supervisors in our allocation mechanism, we introduce a *supervisor utility* term. This additional component aligns the objectives of agents and supervisors within a unified potential framework, ensuring that supervisors prefer assignments with fewer, better-ranked advisees.

6.1 Definition of Supervisor Utility

Let $s_i \in S$ be a supervisor with fixed value $v_i > 0$. Denote by $S_i = \{a_j \in A : x_{i,j} = 1\}$ the set of agents assigned to s_i, and let $\text{rank}_i(a_j)$ be the position of agent a_j in s_i's strict preference list. We define the supervisor utility $W_i(x)$ as

$$W_i(x) \;=\; \frac{v_i}{\sum_{a_j \in S_i} \text{rank}_i(a_j)} \tag{40}$$

Intuitively:

- A smaller denominator (fewer assigned agents or generally lower total rank) increases $W_i(x)$, reflecting a supervisor's preference for light loads.

- A set of well-ranked (highly preferred) agents yields a smaller sum of ranks, further boosting $W_i(x)$.

6.2 Modified Potential Function

Recall the original potential function on agent utilities:

$$\Phi_{\text{old}}(U) \;=\; \sum_{j=1}^{n} \log U_j(x)$$

We extend this to include supervisor utilities, defining

$$\Phi_{\text{new}}(x) \;=\; \Phi_{\text{old}}(U(x)) \;+\; \sum_{i=1}^{k} W_i(x). \tag{41}$$

This augmented potential has several advantages:

1. **Alignment of Incentives**: As agents adjust their assignments to increase U_j, they also contribute indirectly to W_i by respecting supervisors' rank preferences and load constraints.

2. **Supervisor Satisfaction**: Supervisors benefit when they have fewer or more preferred advisees, so the mechanism naturally discourages excessive clustering of agents under a single supervisor.

3. **Local Equilibrium**: Any unilateral deviation by an agent that improves its own utility U_j but worsens one or more W_i may fail to increase the overall Φ_{new}, preventing the move at equilibrium.

6.3 Equilibrium Characterization

Under best-response dynamics, each agent's update increases the global potential Φ_{new}. Since utilities $U_j(x)$ are bounded and each $W_i(x)$ is positive and bounded above by v_i, the process converges to a fixed point where no agent can unilaterally improve Φ_{new}. At convergence:

$$\Phi_{\text{new}}(x) \geq \Phi_{\text{new}}(x') \quad \forall\, x' \text{ reachable by a single-agent deviation.}$$

Equivalently, for every agent a_j considering a move from s_i to $s_{i'}$:

$$U_j(x) + \sum_{i=1}^{k} W_i(x) \;\geq\; U_j(x') + \sum_{i=1}^{k} W_i(x'),$$

ensuring that no unilateral reallocation yields a higher combined agent–supervisor welfare. Hence, the equilibrium satisfies both

$$\begin{aligned} U_j(s_i, s_{-j}) &\geq U_j(s_i', s_{-j}), \quad \forall\, j, s_i', \\ W_i(s_i, s_{-i}) &\geq W_i(s_i', s_{-i}), \quad \forall\, i, s_i', \end{aligned}$$

and can be interpreted as an equilibrium in the joint assignment game.

7 Results

We conducted computational experiments to evaluate the performance of our proposed algorithm against the optimal solution obtained through linear programming (LP). The simulation environment consisted of varying numbers of agents $n \in \{5, 8, 10, 12\}$ and supervisors $k \in \{3, 4, 5\}$, with supervisor values v_i randomly sampled from the uniform distribution $U[36, 60]$. The preference decay parameter γ was varied across $\{0.3, 0.5, 0.7, 0.9\}$ to analyze its impact on envy levels.

The implementation code is available at `https://github.com/utkarshpatel7356/BTP_Paper1`.

7.1 Comparative Performance

Table 1 presents the maximum envy values obtained by our algorithm compared to the optimal LP solution across different problem configurations.

Table 1: Comparison of Maximum Envy Values

Agents	Supervisors	γ	LP Max Envy	Algorithm Max Envy
5	3	0.3	40.10	40.10
8	4	0.3	34.93	41.60
10	4	0.3	33.57	37.36
5	3	0.5	21.28	28.26
8	4	0.5	19.94	19.94
10	4	0.5	26.38	38.32
12	5	0.5	26.17	27.17
5	3	0.7	20.69	24.19
8	4	0.7	11.53	15.52
10	4	0.7	16.37	27.26
12	5	0.7	19.03	28.09
5	3	0.9	18.02	20.20
8	4	0.9	9.86	19.80
10	4	0.9	9.29	11.35
12	5	0.9	10.96	14.34

7.2 Impact of γ on Envy Levels

Figure 1 illustrates how the preference decay parameter γ influences the maximum envy for both the LP optimal solution and our algorithm.

7.3 Key Insights

Our experimental evaluation reveals several important insights:

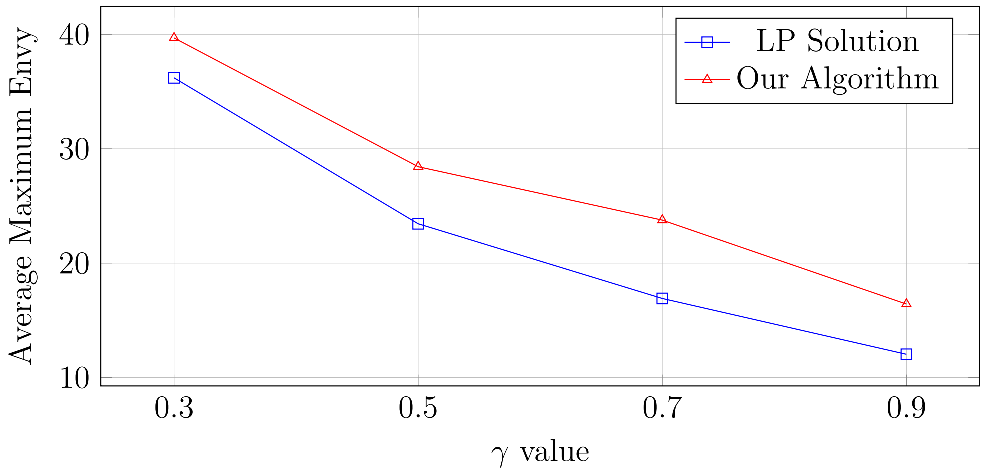

Figure 1: Impact of preference decay parameter γ on average maximum envy

- **Effect of γ:** As γ increases (weaker preference decay), maximum envy decreases substantially for both methods. With $\gamma = 0.3$, the average maximum envy is about three times higher than with $\gamma = 0.9$, indicating that stronger preference distinctions lead to significantly higher envy levels.

- **Performance Gap:** Our algorithm achieves optimal performance in some instances (e.g., $n = 5$, $k = 3$, $\gamma = 0.3$ and $n = 8$, $k = 4$, $\gamma = 0.5$), while showing a moderate performance gap in others. The average gap increases as γ increases, suggesting that our algorithm handles problems with stronger preference distinctions more effectively.

- **Scalability:** The performance gap between our algorithm and the optimal LP solution varies with problem size. For smaller instances ($n = 5$, $k = 3$), the gap remains relatively narrow across all γ values, while for larger instances, particularly with $\gamma = 0.9$, the gap widens more significantly.

- **Computational Efficiency:** While not explicitly measured in our experiments, our algorithm provides solutions in polynomial time compared to the potentially exponential complexity of solving the LP formulation directly, making it particularly valuable for larger problem instances.

These results demonstrate that our algorithm provides an effective approach to the supervisor allocation problem, especially in scenarios with stronger preference distinctions (lower γ values), offering a reasonable approximation to the optimal solution while maintaining computational tractability.

8 Conclusion and Future Work

We presented a novel framework for reward allocation in congestion games that minimizes total envy using an iterated best response algorithm. Agents update their assignments by maximizing a global potential function

$$\Phi(\mathbf{r}) = \sum_{i=1}^{n} \log(U_i),$$

which leads to a local equilibrium and a fair reward distribution.

For future work, we plan to extend this model to a decentralized multi-player multi-armed bandit framework. In this setting, rewards and preference orders are unknown, arms provide stochastic rewards, and collisions result in proportional reward sharing. We also aim to incorporate communication among agents to improve coordination and drive the system toward a stable matching that respects both agent and arm preferences.

In summary, our current work lays a solid foundation for fair reward allocation, while future extensions promise enhanced applicability in dynamic multi-agent environments place.

References

[1] R. Konda, R. Chandan, D. Grimsman, and J. R. Marden, "Best response sequences and tradeoffs in submodular resource allocation games," *arXiv preprint arXiv:2406.17791*, 2024.

[2] M. Fischer, M. Gairing, and D. Paccagnan, "Fair interventions in weighted congestion games," *arXiv preprint arXiv:2311.16760*, 2023.

[3] L. Pedroso, A. Agazzi, W. M. Heemels, and M. Salazar, "Fair artificial currency incentives in repeated weighted congestion games: Equity vs. equality," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pp. 954–959, IEEE, 2024.

[4] K. Nishimura and H. Sumita, "Envy-freeness and maximum nash welfare for mixed divisible and indivisible goods," *arXiv preprint arXiv:2302.13342*, 2023.

[5] H. Ackermann, P. W. Goldberg, V. S. Mirrokni, H. Röglin, and B. Vöcking, "A unified approach to congestion games and two-sided markets," *Internet Mathematics*, vol. 5, no. 4, pp. 439–457, 2008.

[6] S. Liu, X. Lu, M. Suzuki, and T. Walsh, "Mixed fair division: A survey," *Journal of Artificial Intelligence Research*, vol. 80, pp. 1373–1406, 2024.

[7] D. Chakrabarty, A. Mehta, and V. Nagarajan, "Fairness and optimality in congestion games," pp. 52–57, 2005.

[8] Q. Cui, Z. Xiong, M. Fazel, and S. S. Du, "Learning in congestion games with bandit feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 11009–11022, 2022.

[9] J. Bredin, R. T. Maheswaran, C. Imer, T. Başar, D. Kotz, and D. Rus, "A game-theoretic formulation of multi-agent resource allocation," pp. 349–356, 2000.

[10] D. Chakrabarty, A. Mehta, and V. Nagarajan, "Fairness and optimality in congestion games," pp. 52–57, 2005.

[11] P. McGlaughlin and J. Garg, "Improving nash social welfare approximations," *Journal of Artificial Intelligence Research*, vol. 68, pp. 225–245, 2020.

[12] H. Jiang, Q. Cui, Z. Xiong, M. Fazel, and S. S. Du, "Offline congestion games: How feedback type affects data coverage requirement," *arXiv preprint arXiv:2210.13396*, 2022.

[13] J. Chhabra, J. Deshmukh, A. Malavalli, K. Sama, and S. Srinivasa, "Modelling the dynamics of identity and fairness in allocation games," in *International Conference on Principles and Practice of Multi-Agent Systems*, pp. 58–73, Springer, 2024.

[14] H. Charkhgard, K. Keshanian, R. Esmaeilbeigi, and P. Charkhgard, "The magic of nash social welfare in optimization: Do not sum, just multiply!," *The ANZIAM Journal*, vol. 64, no. 2, pp. 119–134, 2022.