Setting Fair Incentives to Maximize Improvement[*]

Saba Ahmadi ✉
Toyota Technological Institute at Chicago, IL, USA

Hedyeh Beyhaghi ✉
Carnegie Mellon University, Pittsburgh, PA, USA

Avrim Blum ✉
Toyota Technological Institute at Chicago, IL, USA

Keziah Naggita ✉
Toyota Technological Institute at Chicago, IL, USA

──── **Abstract** ────

We consider the problem of helping agents improve by setting goals. Given a set of target skill levels, we assume each agent will try to improve from their initial skill level to the closest target level within reach (or do nothing if no target level is within reach). We consider two models: the *common* improvement capacity model, where agents have the same limit on how much they can improve, and the *individualized* improvement capacity model, where agents have individualized limits. Our goal is to optimize the target levels for social welfare and fairness objectives, where *social welfare* is defined as the total amount of improvement, and we consider fairness objectives when the agents belong to different underlying populations. We prove algorithmic, learning, and structural results for each model.

A key technical challenge of this problem is the non-monotonicity of social welfare in the set of target levels, i.e., adding a new target level may decrease the total amount of improvement; agents who previously tried hard to reach a distant target now have a closer target to reach and hence improve less. This especially presents a challenge when considering multiple groups because optimizing target levels in isolation for each group and outputting the union may result in arbitrarily low improvement for a group, failing the fairness objective. Considering these properties, we provide algorithms for optimal and near-optimal improvement for both social welfare and fairness objectives. These algorithmic results work for both the common and individualized improvement capacity models. Furthermore, despite the non-monotonicity property and interference of the target levels, we show a placement of target levels exists that is approximately optimal for the social welfare of each group. Unlike the algorithmic results, this structural statement only holds in the common improvement capacity model, and we illustrate counterexamples of this result in the individualized improvement capacity model. Finally, we extend our algorithms to learning settings where we have only sample access to the initial skill levels of agents.

──────────────

[*] Authors are ordered alphabetically.

1 Introduction

One of the principal practices in policy making is setting reasonable expectations for the groups or individuals involved in the policy. Whether it is in the context of public policy, education, or career development, a too low expectation (i.e., an easy-to-achieve goal) may be a cause for not improving at one's capacity, while a too high expectation (i.e., an out-of-reach goal) may be discouraging to even make an attempt. To accommodate different levels of participants' abilities, the policy-maker may consider different levels of expectations and design goals at various levels so that all (or most of) the participants have a goal within their reach but not too easily achieved. At the other side from the policy-maker are the participants who may view their goal as a mere requirement for accessing other benefits, e.g., increasing their chances of promotion or gaining freedom to pursue other opportunities. In this case, the individuals may choose an easy-to-achieve goal rather than aim for maximal improvement. Examples of such expectations include reading goals for youth, language proficiency goals for applicants, outreach activities for employers, etc.

In this work, we consider the problem of helping agents improve by setting goals. Given a set of target "skill levels", we assume each agent will try to improve from their initial skill level to the closest target level within reach (or do nothing if no target level is within reach). The designer's goal is to maximize the total improvement both with and without fairness considerations.

Mathematically, we formulate this problem as follows. There are n agents belonging to g distinct groups. Agent i has an initial skill level, $p_i \in \mathbb{Z}_{\geq 0}$, and can increase their skill by at most Δ_i which is called its "improvement capacity". Given a set of target levels $\mathcal{T} \subset \mathbb{Z}_{\geq 0}$, agent i improves to the closest target $\tau \in \mathcal{T}$ such that $\tau > p_i$ and $\tau \leq p_i + \Delta_i$ if such target exists; otherwise it stays at p_i.[1]

This problem formulation gives rise to multiple challenges. First, optimizing improvement for a set of agents may conflict with another set. Consider a beginner-level agent (skill level B) and an intermediate-level (skill level I). Agent I finds any level up to τ_I within reach. Therefore, we need to design a project at level τ_I for this agent to improve maximally. On the other hand, B has the capacity to improve until τ_B, where $I < \tau_B < \tau_I$ – See Figure 1a. Now, consider both target levels τ_B and τ_I. Since agent I now has a closer target of τ_B, this agent no longer achieves its maximum improvement, and only reaches skill level τ_B. Secondly, there is non-monotonicity in the placement of target levels, i.e., adding a new target to the current placement may decrease the total amount of improvement. Consider a beginner-level (B) and an intermediate-level (I) agent and a target, τ, achievable by both agents – See Figure 1b. Designing a new project at level τ' between B and τ decreases the total amount of improvement since one agent (if $B < \tau' \leq I$) or both agents (if $I < \tau' < \tau$) switch from improving to τ to improving to τ', which requires less improvement.

(a) Conflict in optimizing improvement.   **(b)** Non-monotonicity in set of target levels.

■ **Figure 1** Challenges in designing optimal target levels.

---

[1] We assume the policy-maker can disallow agent i from choosing a target $\tau \leq p_i$. For example, in designing reading goals for grade-school students, the fifth graders are not allowed to choose materials from the second-grade level, although the reverse is allowed. Setting the base for each agent at their true skill level is an abstraction of our mathematical model.

Main Results. In this work, we consider algorithmic, fairness, and learning-theoretic formulations, where a set of optimal target levels must be found in the presence of effort-bounded agents. We use *social welfare* as the notion of efficiency and define it as the total amount of improvement. Also, we define *social welfare for a given group* as the amount of improvement that group achieves. We consider two models: (1) the common improvement capacity model, where agents have the same limit Δ on how much they can improve, and (2) the individualized improvement capacity model, where agents have individualized limits Δ_i.

The main results of the paper are:

1. An efficient algorithm for placement of target levels to maximize social welfare. (Section 3)
2. An efficient algorithm for outputting the Pareto-optimal outcome for the social welfare of multiple groups. In particular, this can output the max-min fair solution that maximizes the minimum total improvement across groups. (Section 4)
3. A structural result on Pareto-optimal solutions: there exists a placement of target levels that simultaneously is approximately optimal for each group. More explicitly, when there are a constant number of groups, the total improvement for each group is a constant-factor approximation of the maximum improvement that we could provide that group if it were the only group under consideration. **This is our main contribution.** (Section 5)
4. An efficient learning algorithm for near-optimal placement of target levels. (Section 6)

The algorithmic results work for both the common and individualized improvement capacity models. However, the structural result only holds in the common improvement capacity model, and we illustrate examples where achieving any nontrivial fraction of optimal for all groups is not possible in the individualized capacity model.

Related work

Our work broadly falls under two general research areas: social welfare maximization in mechanism design and algorithmic fairness. Specifically, the closest topics to our paper are designing portfolios for consumers to minimize loss of returns [13], designing badges to steer users' behavior [4], and the literature on strategic classification.

Closest to our work is Diana et al.[13] who consider a model where each agent has a risk tolerance, observed as a real number, and must be assigned to a portfolio with risk lower than what they can tolerate. The goal of the mechanism designer is to design a small number of portfolios that minimizes the sum of the differences between the agent's risk tolerance and the risk of the portfolio they take; in other words, it minimizes the loss of returns. Since this is a minimization problem where each agent selects the closest target (portfolio) below their risk tolerance, adding any new target can only help with the objective function. Therefore, unlike our model, there is no conflict between targets, and the objective function is monotone in the set of targets.

Designing targets to incentivize agents to take specific actions is also a common feature of online communities and social media sites. In these platforms, there is a mechanism for rewarding user achievements based on a system of *badges* (similar to targets in our model) [4, 15, 5, 11, 12]. Among such papers, the closest to ours is Anderson et al.[4] who investigate how to optimally place badges in order to induce particular user behaviors, among other things. They consider a dynamic setting with a single user type interested in a particular distribution of actions and a mechanism designer whose objective is to set badges to motivate a different distribution of actions. Compared to our work, their model is more general in the sense that users can spend effort on different actions (improve in multiple dimensions), but also more specific, in the sense that there is only one user type; therefore, unlike our model there is no conflict between different users and adding more badges for the desired action always helps with steering the users in that direction (it is a monotone setting).

Another line of work that is relevant is strategic classification. In most cases, agents are fraudulently strategic, that is to say, game the decision-making model to get desired outcomes (see [18, 14, 20, 23, 1, 10, 16, 9, 26] among others). In other cases, in addition to actions only involving gaming the system, agents can also perform actions that truthfully change themselves to become truly qualified (see [21, 19, 3, 28, 22, 17, 7, 25, 8, 2] among others). In this paper, we assume agents only truthfully change themselves and, therefore, focus on incentivizing agents to improve as much as they can.

Within the combinatorial optimization literature, our work is related to the *uncapacitated facility location* (UFL) problem (see Chapter 4.5 of [27] for the problem definition). The main distinction between our problem and UFL is that the objective in UFL is to minimize the total distance traveled by the clients to reach their closest facility; whereas in our problem, the goal is to maximize the total distance traveled by agents to their closest target within reach.

Organization of the Paper

Section 2 formally introduces the general model settings and definitions used in the paper, and Section 3 provides an efficient algorithm for the problem of maximizing total improvement. Section 4 provides algorithms that output Pareto optimal solutions for groups' social welfare, including a solution that maximizes the minimum improvement per group. Section 5 provides an algorithm that finds the best simultaneously approximately optimal improvement per group and show it provides a constant approximation when the number of groups is constant. In Section 6, we provide efficient learning algorithms which generalize the previous results to a setting where there is only sample access to agents, and Appendix E provides further extensions to our main problems. All missing proofs are deferred to the appendix.

2   Model and Preliminaries

There are n agents $1, \ldots, n$. Agent i is associated with two quantifiers: initial skill level, p_i, and *improvement capacity*, Δ_i, which determines the maximum amount agent i can improve its skill. For the majority of the paper, we assume p_i and Δ_i belong to $\mathbb{Z}_{\geq 0}$; however, some of our results hold more generally for real numbers. We consider two different models. The *common* and the *individualized* improvement capacity models. In the first model, all agents have the same improvement capacity, i.e., Δ_i are equal across agents; we substitute Δ_i with Δ in this case. The second model is a generalization where Δ_i may have different values. We use $\Delta_{max} = \max\{\Delta_1, \cdots, \Delta_n\}$. Our solution is a finite set of target levels $\mathcal{T} \subset \mathbb{Z}_{\geq 0}$. We assume we are given a maximum number of allowed target levels k (if $k = n$, this is equivalent to allowing an unbounded number of target levels).

Agents behavior. Given target levels $\mathcal{T} \subset \mathbb{Z}_{\geq 0}$, agent i aims for the closest target above its initial skill if it can reach to that target given its improvement capacity. More formally, agent i aims for $\min\{\tau \in \mathcal{T} : p_i < \tau \leq p_i + \Delta_i\}$ if such τ exists and improves from p_i to τ. If no such target exists, agent i does not improve and its final skill level remains the same as the initial skill level p_i.

We use *social welfare* (SW) as our notion of efficiency and define it as the total amount of improvement of agents.

Groups and fairness notion. Each agent belongs to one of g distinct groups $G_1, \cdots, G_g$. Given any set of target levels, the social welfare of group ℓ, SW_ℓ, is defined as the total amount of improvement for agents in that group.[2] We are interested in Pareto-optimal solutions for groups' social welfare. A solution $\mathcal{T}$ is Pareto-optimal (is on the Pareto frontier) if there does not exist $\mathcal{T}'$ in which all groups gain at least as much social welfare, and one group gains strictly higher. In particular, the Pareto frontier includes the max-min solution that maximizes the minimum social welfare across groups. In this paper, we focus on two natural fairness notions: one is the max-min solution described above, and the other is the notion of simultaneous approximate optimality given below.

▶ **Definition 1** (Simultaneous α-approximate optimality.). *A solution with at most k targets is simultaneously approximately optimal for each group with approximation factor $0 \leq \alpha \leq 1$ if, for each group ℓ, the social welfare of group ℓ is at least an α fraction of the maximum social welfare achievable for group ℓ using at most k targets.*

Observation 2 determines the potential positions of the targets in an optimal solution.

▶ **Observation 2.** *Without loss of optimality, the targets in an optimal solution are either at positions $p_i + \Delta_i$ or p_i for some $i \in \{1, 2, \ldots, n\}$. Consider a solution where target τ does not satisfy this condition. By shifting τ to the right as long as it does not cross $p_i + \Delta_i$ or p_i for any i, the total amount of improvement weakly increases: This transformation does not change the sets of agents that reach each target, and only increases the improvement of agents aiming for τ.*

▶ **Definition 3** ($\mathcal{T}_p$). *The set of potential optimal target levels, $\mathcal{T}_p$, is $\bigcup_{i=1}^{n} \{p_i, p_i + \Delta_i\}$.*

3 Maximizing Total Improvement

Algorithm 1 provides an efficient dynamic programming algorithm for finding a set of k target levels that maximizes total improvement for a collection of n agents. The recursion function $T(\tau, \kappa)$ finds the best set of at most κ target levels for agents on or to the right of τ. Recall that any target τ only affects the agents on its left, and agent i such that $p_i < \tau$ never selects $\tau' > \tau$ in presence of τ. In the recursive step (item 3 in Algorithm 1), $T(\tau, k)$ is optimized by picking a target $\tau' > \tau$ that maximizes the total improvement achieved by τ' plus the total improvement achieved in the subproblem $T(\tau', \kappa - 1)$.

▶ **Algorithm 1.** *Run dynamic program based on function T, defined below, that takes $\cup_i \{p_i\}$ and k as input and outputs $T(\tau_{\min}, k)$, as the optimal improvement, and $S(\tau_{\min}, k)$, as the optimal set of targets; where $\tau_{\min} = \min\{\tau \in \mathcal{T}_p\}$ and $\tau_{\max} = \max\{\tau \in \mathcal{T}_p\}$. $T(\tau, \kappa)$ captures the maximum improvement possible for agents on or to the right of $\tau \in \mathcal{T}_p$ when at most κ target levels can be selected. Function T is defined as follows.*
1) *For any $\tau \in \mathcal{T}_p$, $T(\tau, 0) = 0$.*
2) *For any $1 \leq \kappa \leq k$, $T(\tau_{\max}, \kappa) = 0$.*
3) *For any $\tau \in \mathcal{T}_p, \tau < \tau_{\max}$ and $1 \leq \kappa \leq k$:*

$$T(\tau, \kappa) = \max_{\tau' \in \mathcal{T}_p \ s.t \ \tau' > \tau} \left(T(\tau', \kappa - 1) \ + \sum_{\tau \leq p_i < \tau' \ s.t. \ \tau' - p_i \leq \Delta_i} (\tau' - p_i) \right)$$

$S(\tau, \kappa)$ keeps track of the optimal set of targets corresponding to $T(\tau, \kappa)$.

---

[2] Although the results are presented for the *total* improvement objective, they also hold for the *average* improvement objective.

▶ **Theorem 4.** *Algorithm 1 finds a set of targets that achieves the optimal social welfare (maximum total improvement) that is feasible using at most k targets given n agents. The algorithm runs in $\mathcal{O}(n^3)$.*

4   Pareto Optimality and Maximizing Minimum Improvement

Algorithm 2 provides a dynamic programming algorithm that constructs the Pareto frontier for groups' social welfare. By iterating through all Pareto-optimal solutions, we can find the solution that maximizes minimum improvement across all groups in pseudo-polynomial time. In contrast to Algorithm 1 where the algorithm only needs to store an optimal solution for each subproblem, here for each subproblem the algorithm stores a set containing *all g*-tuples of groups' improvements $(I_1, I_2, \cdots, I_g)$ that are simultaneously achievable for groups $\{G_1, \cdots, G_g\}$.

▶ **Algorithm 2.** *Run dynamic program based on function T, defined below, that takes $\forall \ell \ \cup_{i \in G_\ell} \{p_i\}$ and k as input and outputs $T(\tau_{\min}, k)$, as the Pareto-frontier improvement tuples, and $S(\tau_{\min}, k)$, as the Pareto-frontier sets of targets ; where $\tau_{\min} = \min\{\tau \in \mathcal{T}_p\}$ and $\tau_{\max} = \max\{\tau \in \mathcal{T}_p\}$. $T(\tau, \kappa)$ constructs the Pareto frontier for groups' social welfare for agents on or to the right of $\tau \in \mathcal{T}_p$ when at most κ target levels can be selected. Function T is defined as follows.*

1) *For any $\tau \in \mathcal{T}_p$, $T(\tau, 0) = \mathbf{0}_g$.*
2) *For any $1 \le \kappa \le k$, $T(\tau_{\max}, \kappa) = \mathbf{0}_g$.*
3) *For any $\tau \in \mathcal{T}_p, \tau < \tau_{\max}$ and $1 \le \kappa \le k$:*

$$T(\tau, \kappa) = \left\{ \left(I_\ell + \left(\sum_{\substack{\tau \le p_i < \tau' \\ s.t. \ \tau' - p_i \le \Delta_i}} \mathbb{1}\left\{ i \in G_\ell \right\}(\tau' - p_i) \right) \right)_{\ell=1}^g , \ s.t. \ (I_\ell)_{\ell=1}^g \in T(\tau', \kappa - 1), \tau' \in \mathcal{T}_p, \tau' > \tau \right\}$$

All the dominated solutions are removed from $T(\tau, \kappa)$. $S(\tau, \kappa)$ stores the sets of targets corresponding to the improvement tuples in $T(\tau, \kappa)$.

▶ **Theorem 5.** *Algorithm 2 constructs the Pareto frontier for groups' social welfare using at most k targets given n agents in g groups. When all p_i, Δ_i values are integral, it has a time-complexity of $\mathcal{O}(n^{g+2}kg\Delta_{\max}^g)$, where $\Delta_{\max}$ is the maximum improvement capacity.*

▶ **Corollary 6.** *There is an efficient algorithm that finds a set of at most k targets that maximizes minimum improvement across all groups, i.e., maximizing $\min_{1 \le \ell \le g} SW_\ell$.*

The algorithm mentioned in Corollary 6 is pseudo-polytime since its time-complexity depends on the numeric value of $\Delta_{\max}$. Appendix C provides a Fully Polynomial Time Approximation Scheme (FPTAS) to maximize the minimum improvement across all groups for the setting where each group G_ℓ has its own improvement capacity Δ_ℓ. In the FPTAS, we assume that p_i, Δ_i values are real numbers.

5   Simultaneous Approximate Optimality

In this section, we establish a structural result about the Pareto optimal solutions in the common improvement capacity model, and show there exists a simultaneously approximately optimal solution on the Pareto frontier, where the approximation factor depends on the number of groups. More specifically, given g groups, and limit $k \ge g$ on the number of target levels, we provide Algorithm 3 whose improvement per group is simultaneously an $\Omega(1/g^3)$ approximation of the optimal k-target solution for each group; implying a constant

approximation when the number of groups is constant. This result is of significance because natural outcomes such as the max-min fair solution and the union of group-optimal targets may lead to arbitrarily poor performance in terms of simultaneous approximate optimality – See Examples 10 and 11. This result only holds for the common improvement capacity model, and in Example 12, we show such a solution does not exist for the individualized improvement capacity model.

▶ **Theorem 7.** *Algorithm 3, given limit $k \geq g$ on the number of target levels, outputs a solution that is simultaneously $\Omega(1/g^3)$-approximately optimal for each group, in the common improvement capacity model. More specifically, it provides a solution such that for all $1 \leq \ell \leq g$, $SW_\ell \geq 1/(16g^3)OPT_\ell^k$, where OPT_ℓ^k is the optimal social welfare of group ℓ using at most k target levels.*

▶ **Corollary 8.** *There is an efficient algorithm to find a simultaneously α^*-approximately optimal solution for each group in the common improvement capacity model, where α^*, defined as the best approximation factor possible, is $\Omega(1/g^3)$.*

We are not aware if $\Omega(1/g^3)$ is the best possible ratio; however, the following example shows there are no simultaneously approximately optimal solutions with approximation factor $> 1/g$.

▶ **Example 9.** Let $\Delta = 1$. Suppose group $\ell \in \{1, 2, \ldots, g\}$ has a single agent at position $(\ell - 1)/g$; i.e., the agents are at $0, 1/g, \ldots, (g-1)/g$. For each group, the optimal total improvement is 1 in isolation (independent of the limit on the number of targets). However, using any number of targets in total there are no solutions with $> 1/g$ improvement for all groups.

The following example shows that the max-min fair solution does not satisfy a simultaneous constant approximation per group even when there are only two groups.

▶ **Example 10.** Let $\Delta = 1$. Group A has n agents; one agent at each position $1, 2, \ldots, n$. Group B has n agents in k bundles of size n/k. The bundles of agents are at positions $n + 1 - k^2/n, \ldots, n + k - k^2/n$. The unique max-min solution has targets at $n - k + 1, n - k + 2, \ldots, n + 1$, and leads to k total improvement for each group which is k/n of the optimal total improvement for group B.

The following example shows solving the optimization problem separately per group and outputting the union of the targets can lead to arbitrarily low group improvement compared to the optimum.

▶ **Example 11.** Suppose there are two groups A and B and no limit on the number of targets. Group A has n agents at positions $1, 3, 5, \ldots, 2n - 1$. Group B has n agents at positions $2 - \varepsilon, 4 - \varepsilon, \ldots, 2n - \varepsilon$. First, consider the common capacity model, where $\Delta = 1$. In this case, the optimal solution for group A in isolation consists of targets at positions $\{2, 4, \ldots, 2n\}$ and the optimal solution for group B is isolation is $\{3 - \varepsilon, 5 - \varepsilon, \ldots, 2n + 1 - \varepsilon\}$. Now, consider a solution that is the union of the targets in the two separate solution. Since each agent in group B is in ε proximity of a target from group A, the total improvement in group B is $n\varepsilon$. Therefore, the total improvement in group B can be arbitrarily close to 0. Next, consider the individualized capacity model, where agents in group A have $\Delta_A = 1$, and agents in group B have $\Delta_B = 1 + 2\varepsilon$. The optimal set of targets in isolation for group A is $\{2, 4, \ldots, 2n\}$, and for group B is $\{3 + \varepsilon, 5 + \varepsilon, \ldots, 2n + 1 + \varepsilon\}$. The union of these solutions result in $1 + (n-1)\varepsilon$ for group A, and $n\varepsilon$ for group B which are arbitrarily low compared to the optimum, which is simultaneously $\geq n(1 - \epsilon)$ for group A and $\geq n$ for group B.

The following example shows that if agents can improve by different amounts (the individualized improvement capacity model), then no approximation factor only as a function of g of optimal improvement per group is possible.

▶ **Example 12.** Suppose groups A and B each have a single agent at position 0. The agent in group A has improvement capacity $\Delta_A = \varepsilon$ and the agent in group B has improvement capacity $\Delta_B = 1$. The optimal total improvement in isolation for group A is ε, and for group B is 1. However, when considering both groups, no placement of targets with positive improvement for group A leads to $> \varepsilon$ improvement for group B.

First, we describe a high-level overview of Algorithm 3 that proceeds in four main steps.
1. **Optimal targets in isolation.** Run Algorithm 1 separately for each group to find an optimal allocation of at most $\lceil k/g \rceil$ targets [3]. Let $\mathcal{T}_\ell$ be the output for group ℓ.
2. **Distant targets in isolation.** Delete $3/4$ fraction of each set of target levels, $\mathcal{T}_\ell$, such that (1) the distance between every two consecutive targets in each set is at least 2Δ and (2) the new $\mathcal{T}_\ell$ (after deletion) guarantees an $\Omega(1)$ approximation of the previous step when the targets for each group are considered in isolation. Section 5.2 below shows this is possible.
3. **Locally optimized distant targets in isolation.** For each ℓ and $\tau \in \mathcal{T}_\ell$, consider the agents in group ℓ that afford to reach τ (agents in $G_\ell \cap [\tau - \Delta, \tau)$). Optimize τ to maximize the total improvement for this set of agents.
4. **Resolve interference of targets.** Consider sets of interfering targets. Relocate these targets locally to guarantee $\Omega(1/g^2)$ approximation per group compared to the previous step where each group was considered in isolation. Section 5.4 below shows this is possible.

5.1 Step 1: Optimal targets in isolation

At the end of step 1, $\mathcal{T}_\ell$ is the optimal set of targets for G_ℓ in isolation. The following observation shows that without loss of optimality, we may assume the distance between every other target level is at least Δ.[4]

▶ **Observation 13.** *Consider a set of target levels $\mathcal{T} : \tau_1 < \tau_2 < \ldots$. Suppose $\tau_{j+2} < \tau_j + \Delta$. By removing τ_{j+1}, any agent with $\tau_j \leq p_i < \tau_{j+1}$ improves strictly more, and other agents improve the same amount. This weakly increases social welfare.*

5.2 Step 2: Distant targets in isolation

Step 2 of the algorithm runs the following procedure for $\mathcal{T}_\ell$.

▶ **Definition 14** (Distant targets procedure). *Consider solution $\mathcal{T} : \tau_1 < \tau_2 < \ldots$, where for all j, $\tau_{j+2} - \tau_j \geq \Delta$ as input to the following procedure.*
- *Partition $\mathcal{T}$ into 4 parts, P_1, P_2, P_3, P_4, where $P_i =: \tau_i, \tau_{4+i}, \tau_{8+i}, \ldots$. Consider the part P_i that introduces the highest improvement. Update $\mathcal{T}$ to P_i (and delete the rest).*

Lemma 15 shows that at the end of this step, target levels in $\mathcal{T}_\ell$ are 2Δ apart, this step provides a 4-approximation compared to the previous step, and the number of targets designated to each group is at most $\lfloor k/g \rfloor$.

---

[3] Although the total number of targets used in this step can be more than k, after the algorithm ends at most k targets are being used in total.
[4] Example 30, however, shows the distance between two *consecutive* targets may be arbitrarily smaller than Δ.

■ **Algorithm 3** Simultaneous approximate optimality per group.

---

1 for $\ell = 1$ *to* g **do**
   /* Step 1 */
2   Let $\mathcal{T}_\ell : \tau_1 < \tau_2 < \ldots$ be the output of Algorithm 1 for agents in G_ℓ and limit
      $\lceil k/g \rceil$ on the number of targets.
   /* Step 2 */
3   Partition $\mathcal{T}_\ell$ to 4 parts P_1, P_2, P_3, P_4, where $P_i := \tau_i, \tau_{4+i}, \tau_{8+i}, \ldots$.
4   Update $\mathcal{T}_\ell$ by keeping the part with the highest improvement and deleting the
      rest.
   /* Step 3 */
5   Update G_ℓ by deleting the agents that do not improve given $\mathcal{T}_\ell$.
6   For all $\tau \in \mathcal{T}_\ell$, replace τ with the output of Algorithm 1 for agents in
      $[\tau - \Delta, \tau) \cap G_\ell$ and limit 1 on the number of targets.
   /* Step 4 */
7 $\mathcal{T} : \tau_1 < \tau_2 < \ldots = \cup_\ell \mathcal{T}_\ell$
8 $S, \mathcal{T}^* = \emptyset$
9 for $\tau_j \in \mathcal{T}$ **do**
10 $s_j = \tau_j - \Delta$
11 $S = S \cup \{s_j\}$
12 Partition $S : s_1 < s_2 < \ldots$ into the least number of parts of consecutive points:
   $S_1, S_2, \ldots$, such that in each part, S_i, each two consecutive points are at distance less
   than Δ/g.
13 for *all* $S_i : s_u < s_{u+1} < \ldots < s_v$ **do**
14 $\tau_i^* = \min\{\tau_u, s_{v+1}\}$.
15 $\mathcal{T}^* = \mathcal{T}^* \cup \tau_i^*$.
16 return $\mathcal{T}^*$

▶ **Lemma 15.** *Consider solution* $\mathcal{T} : \tau_1 < \tau_2 < \ldots$ *with total improvement* I *such that for all* j, $\tau_{j+2} - \tau_j \geq \Delta$. *Consider the procedure in Definition 14. This procedure results in a solution* $\mathcal{T}' : \tau_1' < \tau_2' < \ldots$ *where* $\forall j \ \tau_{j+1}' - \tau_j' \geq 2\Delta$, *has total improvement at least* $I/4$, *and* $|\mathcal{T}'| \leq \lceil |\mathcal{T}|/4 \rceil$. *Particularly, for* $|\mathcal{T}| \leq \lceil k/g \rceil$ *where* $k \geq g$, *the number of final targets,* $|\mathcal{T}'|$, *is at most* $\lfloor k/g \rfloor$.

Proof. Since the best out of 4 parts have been selected, the total improvement at the end of the procedure is at least $1/4$ fraction of I. In addition, in the final set, every pair of consecutive targets are indexed τ_j and τ_{j+4}. Therefore, since originally for all j, $\tau_{j+2} - \tau_j \geq \Delta$, we have $\tau_{j+4} - \tau_j \geq 2\Delta$. Finally, since in each set of $\tau_j, \ldots, \tau_{j+4}$ exactly one target is selected, the final number of targets is at most $\lceil |\mathcal{T}|/4 \rceil$. ◀

5.3 Step 3: Locally optimized distant targets in isolation

At the end of step 2, every two targets in $\mathcal{T}_\ell$, the set of targets for group ℓ, are at distance at least 2Δ. Consider only the targets and agents in group ℓ. For each $\tau \in \mathcal{T}_\ell$, agents in $[\tau - \Delta, \tau)$ improve to τ and the remaining agents do not improve. To continue with the algorithm, we first delete the agents that do not improve (and update G_ℓ accordingly). Then, we optimize $\mathcal{T}_\ell$ for the set of agents that do improve. This modification is necessary for the next step. To do the optimization, we use Algorithm 1 for agents in $[\tau - \Delta, \tau)$ for any $\tau \in \mathcal{T}_\ell$ and limit 1 on the number of targets, and replace τ with the output of the algorithm.

▶ **Lemma 16.** *At the end of step* 3 *in Algorithm 3, (i) the distance between every two targets in* $\mathcal{T}_\ell$ *is at least* Δ*; (ii) each target* $\tau \in \mathcal{T}_\ell$ *is optimal, i.e., maximizes total improvement for the remaining agents in* $G_\ell \cap [\tau - \Delta, \tau)$*; and (iii) the total amount of improvement of* G_ℓ *using solution* $\mathcal{T}_\ell$ *does not decrease compared to the previous step.*

Proof. Let τ be a target at the beginning of step 3 and τ' be its replacement at the end of this step.

We first prove statement (i). First, we argue for agents in $[\tau - \Delta, \tau)$, the optimal target τ' belongs to $[\tau, \tau + \Delta]$. Intuitively, the reason is that all these agents afford to improve to τ; therefore, a target smaller than τ is suboptimal. Also, none of the agents affords to improve beyond $\tau + \Delta$. More formally, if $\tau' < \tau$, agents in $[\tau - \Delta, \tau')$ improve less compared to a target at τ and agents in $[\tau', \tau)$ do not improve. On the other hand, if $\tau' > \tau + \Delta$, none of the agents can reach τ' and the total improvement for these agents will be 0. Therefore, at the end of this step, every target τ is replaced with $\tau' \in [\tau, \tau + \Delta]$. Now, by Observation 13 and Lemma 15, the distance between consecutive targets at the end of step 2 is at least 2Δ. Therefore, after the modification explained (shifting each target to the right by less than Δ) this distance decreases by at most Δ and becomes at least Δ.

Now, we move on to statement (ii). We need to argue if τ' is optimal for agents in $[\tau - \Delta, \tau)$, it is also optimal for agents in $[\tau' - \Delta, \tau')$. By Lemma 15, at the beginning of step 3, there are no targets in $(\tau, \tau + 2\Delta)$; more specifically, there are no targets for agents in $[\tau, \tau + \Delta)$ and these agents get eliminated in this step. Therefore, since τ' belongs to $[\tau, \tau + \Delta]$, as shown in the proof of statement (i), we only need to argue that if τ' is optimal for $[\tau - \Delta, \tau)$, it is also optimal for $[\tau' - \Delta, \tau)$. Suppose this was not the case, and there was another target τ'' which was optimal for this set. Since the agents in $[\tau' - \Delta, \tau)$ are the only agents with positive amount of improvement for target τ', replacing τ' with τ'' would result in higher improvement for the whole set of agents in $[\tau - \Delta, \tau)$ which is in contradiction with definition of τ'.

Finally, we argue statement (iii). In step 3, we consider two sets of agents: those who do not improve in step 2, and those who do. The new targets in this step have been optimized for the second set and (weakly) increase their total improvement. Since the first set did not have any improvement in the first place, the total amount of improvement (for the first and second set) does not decrease in this step. ◀

Next, we extract properties about optimal solutions. Since at the end of step 3, $\mathcal{T}_\ell$ is optimal for G_ℓ we take advantage of these properties in the remaining steps of the algorithm. Lemma 17 shows that if τ is optimal for agents in $[\tau - \Delta, \tau)$, a considerable fraction of these agents reside in the left-most part of the interval. Lemma 18 shows that if τ is optimal for agents in $[\tau - \Delta, \tau)$, substituting τ with another target in this interval, far enough from the left endpoint, $\tau - \Delta$, guarantees a considerable fraction of the optimal improvement.

▶ **Lemma 17.** *Consider optimal target* τ *for the set of agents* A *in* $[\tau - \Delta, \tau)$ *in absence of other targets. For each* $0 \leq x \leq 1$*, at least* x *fraction of* A *belong to* $[\tau - \Delta, \tau - \Delta + x\Delta)$*. In particular, at least* $1/(2g)$ *fraction of the agents are in* $[\tau - \Delta, \tau - (2g - 1)/(2g)\Delta)$*.*

Proof. Let p_x be the fraction of agents in A in $[\tau - \Delta, \tau - \Delta + x\Delta)$. Each of these agents is improving by at least $(1 - x)\Delta$. Therefore, the contribution of these agents to total improvement of A is at least $p_x|A|(1 - x)\Delta$. Since τ is the optimal target, it introduces at least as much improvement as any other target, and in particular a target at $\tau' = \tau + x$. Consider the total improvement introduced by τ' compared to τ (in absence of target τ). The contribution of the agents in $[\tau - \Delta, \tau - \Delta + x\Delta)$ to total improvement reduces to 0, but

the contribution of the agents in $[\tau - \Delta + x\Delta, \tau)$ increases by $(1 - p_x)|A|x\Delta$. Since τ is the optimal target, the loss of substituting it with τ' is at least as much as the gain. Therefore, $p_x(1 - x)\Delta \geq (1 - p_x)x\Delta$; which implies $p_x \geq x$. ◄

▶ **Lemma 18.** *Consider optimal target τ for agents A in $[\tau - \Delta, \tau)$ in absence of other targets. By relocating τ to any point in $[\tau - \Delta + x\Delta, \tau]$, for $0 \leq x \leq 1$, the total improvement of A is at least $x^2/4$ of the optimum. In particular, by relocating τ to any point in $[\tau - \Delta + \Delta/g, \tau]$, the total improvement is at least $1/(4g^2)$ of the optimum.*

Proof. Similar to the previous lemma, let $p_{x/2}$ be the fraction of agents in $[\tau - \Delta, \tau - \Delta + (x/2)\Delta)$. After the relocation, each such agent improves by at least $(x/2)\Delta$; therefore, the contribution of these agents to total improvement is at least $p_{x/2}|A|(x/2)\Delta$. The optimal total improvement is bounded by $|A|\Delta$. Therefore, using $p_{x/2} \geq x/2$, by Lemma 17, the total improvement after relocation is at least $x^2/4$ of the optimum. ◄

5.4 Step 4: Resolve interference of targets

In this step, we consider the solutions for all groups together and resolve the interference of targets designed for different groups. As illustrated in Example 11, this interference can lead to arbitrarily low social welfare. To resolve this issue, we take advantage of sparsity of the targets designed for the same group (step 2) and optimality of $\mathcal{T}_\ell$ for G_ℓ (step 3).

The main purpose of this step is to recover an approximation guarantee of the total improvement of *each target in isolation* at the end of step 3 by removing the interference among the targets. Particularly, for each target $\tau \in \mathcal{T}_\ell$ in isolation, we consider agents in G_ℓ reaching to that, i.e., agents in interval $[\tau - \Delta, \tau)$. By Lemma 17, a considerable fraction of these agents are on the left-most side of the interval. And as shown in Lemma 18, as long as there exists a target far enough from the left endpoint we are in good shape. More precisely, if for all τ at the beginning of this step, there is a target in the final solution in $[\tau - \Delta + \Delta/g, \tau]$ (property 1), and no targets in $(\tau - \Delta, \tau - \Delta + \Delta/g)$ (property 2), a $1/(4g^2)$ fraction is achievable. The set of targets at the end of step 3 may fail to satisfy these properties, because there may be targets $\tau' < \tau$ such that τ' is not far enough from the left endpoint of the interval corresponding to τ; i.e., for $s = \tau - \Delta$, $s < \tau' < s + \Delta/g$.

To resolve the interference among the targets, in step 4, we work as follows. First, we consider the left endpoints of improvement intervals corresponding to the targets; i.e., $\forall \tau_j$, at the end of step 3, consider $s_j = \tau_j - \Delta$. Then, we partition these left endpoints into maximal parts $S_1, S_2, \ldots$, such that in each part, the distance between every two consecutive points is small, particularly, less than Δ/g. Using the sparsity of the targets (step 2) the number of points in each part is bounded. Finally, we design a new target τ_i^* (defined formally below) corresponding to part S_i, such that τ_i^* is to the left of any S_j with $j > i$, and at distance between Δ/g and Δ to the right of the points in S_i (satisfying properties 1 and 2). Using optimality of $\mathcal{T}_\ell$ for G_ℓ (step 3) this results in the desired approximation factor. More formally, this step proceeds as follows.

1. Let $\mathcal{T} : \tau_1 < \tau_2 < \ldots$ be the union of the set of targets found at the end of step 3.
2. Construct $S : s_1 < s_2 < \ldots$ from $\mathcal{T}$, such that $\forall \tau_j \in \mathcal{T}$, include $s_j = \tau_j - \Delta$ in S.
3. Partition S into the least number of parts of consecutive points: $S_1, S_2, \ldots$, such that in each part $S_i : s_u < s_{u+1} < \ldots < s_v$, each two consecutive points are at distance less than Δ/g; i.e., $\forall s_r, s_{r+1} \in S_i, s_{r+1} - s_r < \Delta/g$. By construction of the first three steps (and as shown in the proof of Lemma 19), the number of points in each part is at most g.
4. For each $S_i : s_u < s_{u+1} < \ldots < s_v$, consider new target $\tau_i^* = \min\{\tau_u, s_{v+1}\}$.
5. Output the set of new targets.

▶ **Lemma 19.** *Consider $\mathcal{T}$ as the union of all solutions at the end of step 3. For all $\tau \in \mathcal{T}$, consider the interval $[\tau - \Delta, \tau)$ which consists of agents that improve to target τ if it were the only target available. At the end of step 4, (i) there will be a target in $[\tau - \Delta + \Delta/g, \tau]$, and (ii) there will be no targets in $(\tau - \Delta, \tau - \Delta + \Delta/g)$.*

Proof. Statement (i) is equivalent to (i') for any $s \in S$, there will be a target in $[s + \Delta/g, s + \Delta]$; and statement (ii) is equivalent to (ii') for any $s \in S$, there will be no targets in $(s, s + \Delta/g)$. We prove (i') and (ii').

We first show the size of each part is at most g; i.e. $\forall i, |S_i| \geq g$. The proof is by contradiction. Suppose there exists $|S_i| \geq g + 1$. Therefore, there exist $s_j < s_{j'} \in S_i$ and group index ℓ, such that $s_j + \Delta, s_{j'} + \Delta \in \mathcal{T}_\ell$, and all s satisfying $s_j < s < s_{j'} \in S_i$ corresponding to targets in distinct groups other than ℓ. Therefore, there are at most $g - 1$ such s. Hence, $s_{j'} - s_j < g \times \Delta/g = \Delta$, implying there are two targets in $\mathcal{T}_\ell$ at distance strictly less than Δ which is in contradiction with Lemma 16.

Now, we prove statement (i''). In step 4, the final target corresponding to part $S_i : s_u \leq s_{u+1} \leq \ldots \leq s_v$ is defined as $\tau_i^* = \min\{\tau_v, s_{v+1}\}$. By definition, $\tau_i^* \leq s_{v+1}$; therefore, it is (weakly) to the left of any s_j for $j \geq v + 1$. Also, using $|S_i| \leq g$, $s_v < s_u + (g-1)\Delta/g$, which implies $\tau_u - s_v > \Delta/g$, and since by definition, $s_{v+1} - s_v \geq \Delta/g$, both s_{v+1} and τ_u are at least at distance Δ/g to the right of s_v and any s_j such that $j \leq v$. This proves statement (i'').

Finally, we prove (i'). In the proof of (ii'), we showed that $\tau_i^* \geq s_v + \Delta/g$ which implies $\tau_i^* \geq s + \Delta/g$, $\forall s \in S_i$. Therefore, it suffices to show $\tau_i^* \leq s_u + \Delta$, which then implies $\tau_i^* \leq s + \Delta$, $\forall s \in S_i$. The definition of τ_i^* directly implies $\tau_i^* \leq s_u + \Delta$. ◄

5.5 Putting everything together

▶ **Theorem 20.** *Algorithm 3, given $k \geq g$, provides a solution with at most k number of targets, such that for all $1 \leq \ell \leq g$, $SW_\ell \geq 1/(16g^2)OPT_\ell^{\lceil k/g \rceil}$, where OPT_ℓ^k is the optimal social welfare of group ℓ using at most k target levels. This statement holds in the common improvement capacity model.*

Proof. By Observation 13 and Lemma 15, when the targets designed for each group are considered separately and in isolation, at the end of step 2, there are at most $\lfloor k/g \rfloor$ targets designed for group ℓ and the total improvement in this group is $1/4$-approximation of $OPT_\ell^{\lceil k/g \rceil}$. By Lemma 16, Lemma 18, and Lemma 19, we lose another $4g^2$ factor compared to step 2. In total, Algorithm 3 results in $SW_\ell \geq 1/(16g^2)OPT_\ell^{\lceil k/g \rceil}$, for all groups $1 \leq \ell \leq g$. Also, when $k \geq g$, the total number of targets is at most $g\lfloor k/g \rfloor \leq k$. ◄

Proof of Theorem 7. Given Theorem 20, it suffices to argue $OPT_\ell^{\lceil k/g \rceil} \geq OPT_\ell^k/g$; i.e., when the number of targets increases by a factor, here g, the optimal total improvement increases by at most that factor. The argument follows using the subadditivity of total improvement as a function of the set of targets. Specifically, consider the optimal k-target solution and an arbitrary partition with g parts of size $\lceil k/g \rceil$ or $\lfloor k/g \rfloor$; by subadditivity, one of the parts provides at least $1/g$ of the total improvement. ◄

Proof of Corollary 8. Algorithm 2 in Section 4 outputs the Pareto frontier for groups' social welfare. By definition, the solution provided in Algorithm 3 is dominated by a solution on the Pareto frontier. By computing the factor of simultaneous approximate optimality of each solution on the Pareto frontier, we find the solution that achieves the best simultaneous approximation factor α^*, and by Theorem 7, this solution is simultaneously $\Omega(1/g^3)$-approximately optimal. ◄

▶ **Remark 21** (a weaker benchmark and a tighter gap). In contrast with Theorem 7 that measures the performance of Algorithm 3 with respect to the optimal k-target solution for each group (the notion of simultaneous approximate optimality), Theorem 20 measures the performance with respect to the optimal $\lceil k/g \rceil$-target solution for each group. Since the lower bound provided in Example 9 shows achieving better than $1/g$ of either of these benchmarks is not possible, there is only a factor g gap in the performance of the algorithm and the lower bound with respect to the optimal $\lceil k/g \rceil$-target solution.

6 Generalization Guarantees

In this section, we generalize our results to a setting where we only have sample access to agents and provide sample complexity results. Section 6.1 provides a guarantee for the maximization objective in absence of fairness, and Section 6.1 provides a guarantee for the fairness objectives.

6.1 Generalization Guarantees For the Maximization Objective

Suppose there is a distribution $\mathcal{D}$ over agents' positions. Our goal is to find a set of k targets $\mathcal{T}$ that maximizes expected improvement of an agent when we only have access to n agents sampled from $\mathcal{D}$. For any distribution $\mathcal{D}$ over agents' positions, we define $I_{\mathcal{D}}(\mathcal{T}) = \mathbb{E}_{p \sim \mathcal{D}}[I_p(\mathcal{T})]$, where $I_p(\mathcal{T})$ captures the improvement of agent p given the targets in $\mathcal{T}$. In Theorem 22, we provide a generalization guarantee that shows if we sample a set S of size $n \geq \varepsilon^{-2}\big(\Delta_{\max}^2(k\ln(k) + \ln(1/\delta))\big)$ drawn *i.i.d* from $\mathcal{D}$, then with probability at least $1 - \delta$, for all sets $\mathcal{T}$ of k targets, we can bound the difference between average performance over S and actual expected performance, such that $\big|I_S(\mathcal{T}) - I_{\mathcal{D}}(\mathcal{T})\big| \leq \mathcal{O}(\varepsilon)$. Formally, we show the following theorem holds:

▶ **Theorem 22** (Generalization of the maximization objective). *Let $\mathcal{D}$ be a distribution over agents' positions. For any $\varepsilon > 0$, $\delta > 0$, and number of targets k, if $S = \{p_i\}_{i=1}^{n}$ is drawn i.i.d. from $\mathcal{D}$ where $n \geq \varepsilon^{-2}\Delta_{\max}^2\big(k\ln(k) + \ln(1/\delta)\big)$, then with probability at least $1 - \delta$, for all sets $\mathcal{T}$ of k targets, $\big|I_S(\mathcal{T}) - I_{\mathcal{D}}(\mathcal{T})\big| \leq \mathcal{O}(\varepsilon)$.*

In particular, the solution $\mathcal{T}^*$ that maximizes improvement on $\mathcal{S}$, also maximizes improvement on $\mathcal{D}$ within an additive factor of $\mathcal{O}(\varepsilon)$.

In order to prove Theorem 22, we use two main ideas. First, using a framework developed by Balcan et al. [6], we bound the *pseudo-dimension* complexity of our improvement function. Then, using classic results from learning theory [24], we show how to translate *pseudo-dimension* bounds into generalization guarantees. The framework proposed by Balcan et al. [6] depends on the relationship between primal and dual functions. When the dual function is piece-wise constant, piece-wise linear or generally piece-wise structured, they show a general theorem that bounds the *pseudo-dimension* of the primal function. Formally *pseudo-dimension* is defined as following:

▶ **Definition 23** (Pollard's Pseudo-Dimension). *A class $\mathcal{F}$ of real-valued functions P-shatters a set of points $\mathcal{X} = \{x_1, x_2, \cdots, x_n\}$ if there exists a set of thresholds $\gamma_1, \gamma_2, \cdots, \gamma_n$ such that for every subset $T \subseteq \mathcal{X}$, there exists a function $f_T \in \mathcal{F}$ such that $f_T(x_i) \geq \gamma_i$ if and only if $x_i \in T$. In other words, all 2^n possible above/below patterns are achievable for targets $\gamma_1, \cdots, \gamma_n$. The pseudo-dimension of $\mathcal{F}$, denoted by $\mathrm{PDim}(\mathcal{F})$, is the size of the largest set of points that it P-shatters.*

Balcan et al. [6] show when the dual function is piece-wise structured, the *pseudo-dimension* of the primal function gets bounded as following:

▶ **Theorem 24** (Bounding Pseudo-Dimension [6]). *Let $\mathcal{U} = \{u_\rho \mid \rho \in \mathcal{P} \subseteq \mathbb{R}^d\}$ be a class of utility functions defined over a d-dimensional parameter space. Suppose the dual class $\mathcal{U}^*$ is $(\mathcal{F}, \mathcal{G}, m)$-piecewise decomposable, where the boundary functions $\mathcal{G} = \{f_{\mathbf{a},\theta} : \mathcal{U} \to \{0,1\} \mid \mathbf{a} \in \mathbb{R}^d, \theta \in \mathbb{R}\}$ are halfspace indicator functions $g_{\mathbf{a},\theta} : u_\rho \to \mathbb{I}_{\mathbf{a}\cdot\rho \leq \theta}$ and the piece functions $\mathcal{F} = \{f_{\mathbf{a},\theta} : \mathcal{U} \to \mathbb{R} \mid \mathbf{a} \in \mathbb{R}^d, \theta \in \mathbb{R}\}$ are linear functions $f_{\mathbf{a},\theta} : u_\rho \to \mathbf{a} \cdot \rho + \theta$, and m shows the number of boundary functions. Then, $\mathrm{PDim}(\mathcal{U}) = \mathcal{O}(d \ln(dm))$.*

We use Theorem 24 to bound the *pseudo-dimension* of the improvement function.

▶ **Lemma 25.** *Let $\mathcal{U} = \{u_\mathcal{T} : p \to u_\mathcal{T}(p) \mid \mathcal{T} \in \mathbb{R}^k, p \in \mathbb{R}\}$ be a set of functions, where each function defined by a set of k targets, takes as input a point $p \in \mathbb{R}$ that captures an agent's position, and outputs a number showing the improvement that the agent can make. Then, $\mathrm{PDim}(\mathcal{U}) = \mathcal{O}(k \ln(k))$.*

Proof. We use Theorem 24 to bound $\mathrm{PDim}(\mathcal{U})$. First, we define the dual class of $\mathcal{U}$ denoted by $\mathcal{U}^*$. The function class $\mathcal{U}^* = \{u_p^* : \mathcal{T} \to u_p(\mathcal{T}) \mid \mathcal{T} \in \mathbb{R}^k, p \in \mathbb{R}\}$ is a set of functions, where each function defined by an agent p, takes as input a set $\mathcal{T} \in \mathbb{R}^k$ of k targets [5], and outputs the improvement that p can make given $\mathcal{T}$. Geometrically, in the dual space, there are k dimensions $\tau_1, \cdots, \tau_k$, and each dimension is corresponding to one target. In order to use Theorem 24, we show that $\mathcal{U}^* = (\mathcal{F}, \mathcal{G}, k)$ is piecewise-structured. The boundary functions in $\mathcal{G}$ are defined as follows. If agent p improves to a target τ_i, then $0 < \tau_i - p \leq \Delta$, where Δ is the improvement capacity of p. Additionally, between all the targets within a distance of at most Δ, p improves to the closest one. For each pair of integers (i, j), where $1 \leq i, j \leq k$, we add the hyperplane $\tau_i - \tau_j = 0$ to $\mathcal{G}$. Above this hyperplane is the region where $\tau_i > \tau_j$, implying that τ_i comes after τ_j. Below the hyperplane is the region where the ordering is reversed. In addition, for each target τ_i, we add the boundary functions $\tau_i = p$ and $\tau_i = p + \Delta$ to $\mathcal{G}$. In the region between $\tau_i = p$ and $\tau_i = p + \Delta$, τ_i is effective and the agent can improve to it. Now, the dual space is partitioned into a set of regions. In each region, either there exists a unique closest effective target (τ_r), or all the targets are ineffective. In the former case, the improvement that the agent makes is a linear function of its distance from the closest effective target $(f = \tau_r - p)$. In the later case, the agent makes no improvement $(f = 0)$. Therefore, the piece functions in $\mathcal{F}$ are either constant or linear. Now, since the total number of boundary functions is $m = \mathcal{O}(k^2)$ and the space is k-dimensional, using Theorem 24, $\mathrm{PDim}(\mathcal{U})$ is $\mathcal{O}(k \ln(k^3)) = \mathcal{O}(k \ln(k))$.   ◀

Now, we are ready to prove Theorem 22.

Proof. Classic results from learning theory [24] show the following generalization guarantees: Suppose $[0, H]$ is the range of functions in hypothesis class $\mathcal{H}$. For any $\delta \in (0, 1)$, and any distribution $\mathcal{D}$ over $\mathcal{X}$, with probability $1 - \delta$ over the draw of $\mathcal{S} \sim \mathcal{D}^n$, for all functions $h \in \mathcal{H}$, the difference between the average value of h over $\mathcal{S}$ and its expected value gets bounded as follows:

$$\left| \frac{1}{n} \sum_{x \in \mathcal{S}} h(x) - \mathbb{E}_{y \sim \mathcal{D}}[h(y)] \right| = \mathcal{O}\left(H \sqrt{\frac{1}{n} \left(\mathrm{PDim}(\mathcal{H}) + \ln(\frac{1}{\delta}) \right)} \right)$$

---

[5] If the input consists of k' targets where $k' < k$, it resembles the case where k targets are used and $k - k'$ of them are ineffective, i.e., are put at position $\tau_{\min}$.

In the case of maximizing improvement, $H = \Delta_{max}$ and $\text{PDim}(\mathcal{H}) = \mathcal{O}(k \ln(k))$. By setting $n \geq \varepsilon^{-2}\Delta_{max}^2\big(k\ln(k)+\ln(1/\delta)\big)$, with probability at least $1-\delta$, the difference between the average performance over $\mathcal{S}$ and the expected performance on $\mathcal{D}$ gets upper-bounded by $\mathcal{O}(\varepsilon)$. ◀

6.2 Generalization Guarantees For Fairness Objectives

Suppose there is a distribution $\mathcal{D}_\ell$ of agents' positions for each group ℓ. Let $\mathcal{D} = \sum_{\ell=1}^g \alpha_\ell \mathcal{D}_\ell$ be a weighted mixture of distributions $\mathcal{D}_1, \cdots, \mathcal{D}_g$. Let $\alpha_{\min} = \min_{1 \leq \ell \leq g} \alpha_\ell$. Suppose we have sampling access to $\mathcal{D}$ and cannot directly sample from $\mathcal{D}_1, \cdots, \mathcal{D}_g$. Our goal is to derive generalization guarantees for different objective functions across multiple groups when we only have access to a set S of n agents sampled from distribution $\mathcal{D}$. Let $I_{G_\ell}(\mathcal{T})$ denote the average improvement of agents in group $G_\ell \subseteq S$ given a set $\mathcal{T}$ of k targets. Let $I_{\mathcal{D}_\ell}(\mathcal{T}) = \mathbb{E}_{p \sim \mathcal{D}_\ell}[I_p(\mathcal{T})]$, where $I_p(\mathcal{T})$ captures the improvement of agent p given $\mathcal{T}$. In Theorem 26, we show if we sample a set S of $\mathcal{O}\Big(\alpha_{\min}^{-1}\Big(\varepsilon^{-2}\Delta_{\max}^2\big(k\ln(k) + \ln(g/\delta)\big) + \ln(g/\delta)\Big)\Big)$ examples drawn *i.i.d.* from $\mathcal{D}$, then for all sets $\mathcal{T}$ of k targets and for all groups ℓ, $\big|I_{G_\ell}(\mathcal{T}) - I_{\mathcal{D}_\ell}(\mathcal{T})\big| \leq \mathcal{O}(\varepsilon)$.

▶ **Theorem 26** (Generalization across multiple groups). *Let $\mathcal{D}$ be a distribution over agents' positions. For any $\varepsilon > 0$, $\delta > 0$, and number of targets k, if $S = \{p_i\}_{i=1}^n$ consisting of g groups $\{G_\ell\}_{\ell=1}^g$ is drawn i.i.d. from $\mathcal{D}$, where $n \geq (2/\alpha_{\min})\big(\varepsilon^{-2}\Delta_{\max}^2(k\ln(k) + \ln(2g/\delta)) + 4\ln(2g/\delta)\big)$, then with probability at least $1 - \delta$, for all sets $\mathcal{T}$ of k targets, for all groups ℓ, $\big|I_{G_\ell}(\mathcal{T}) - I_{\mathcal{D}_\ell}(\mathcal{T})\big| \leq \mathcal{O}(\varepsilon)$.*

Proof. Let S be partitioned into g groups where each group G_ℓ has size n_ℓ. First, for each group ℓ, let A_ℓ denote the event that $n_\ell \geq (n\alpha_\ell)/2$. Using Chernoff-Hoeffding bounds we have $\Pr[n_\ell < (n\alpha_\ell)/2] \leq e^{(-n\alpha_\ell)/8} \leq \delta/(2g)$. The last inequality holds since $n \geq 8\ln(2g/\delta)/\alpha_\ell$. Next, for each group ℓ, let B_ℓ denote the event that $\big|I_{G_\ell}(\mathcal{T}) - I_{\mathcal{D}_\ell}(\mathcal{T})\big| \leq \mathcal{O}(\varepsilon)$, then:

$$\Pr[B_\ell] \geq \Pr[B_\ell \cap A_\ell] = \Pr[B_\ell \mid A_\ell] \cdot \Pr[A_\ell] \geq (1 - \delta/(2g))(1 - \delta/(2g)) \geq (1 - \delta/g) \quad (1)$$

In the above statement, inequality $\Pr[B_\ell \mid A_\ell] \geq (1 - \delta/(2g))$ holds since given A_ℓ happens, then $n_\ell \geq \varepsilon^{-2}\Delta_{max}^2(k\ln(k) + \ln(2g/\delta))$, and by Theorem 22, event B_ℓ happens with probability at least $1 - \delta/(2g)$. Now, by Equation (1), $\Pr[B_\ell] \geq 1 - \delta/g$. By applying a union bound, event B_ℓ happens with probability at least $1 - \delta$ for any group ℓ. ◀

In particular, solution $\mathcal{T}^*$ satisfying one of the fairness notions considered in this paper, e.g., simultaneous approximate optimality or maxmizing minimum improvement across groups, on input S, achieves a performance guarantee within an additive factor of $\mathcal{O}(\varepsilon)$ on inputs drawn from $\mathcal{D}$.

────── **References** ──────

1   Saba Ahmadi, Hedyeh Beyhaghi, Avrim Blum, and Keziah Naggita. The strategic perceptron. In Péter Biró, Shuchi Chawla, and Federico Echenique, editors, *EC '21: The 22nd ACM Conference on Economics and Computation, Budapest, Hungary, July 18-23, 2021*, pages 6–25. ACM, 2021. `doi:10.1145/3465456.3467629`.

2   Saba Ahmadi, Hedyeh Beyhaghi, Avrim Blum, and Keziah Naggita. On classification of strategic agents who can both game and improve. In L. Elisa Celis, editor, *3rd Symposium on Foundations of Responsible Computing, FORC 2022, June 6-8, 2022, Cambridge, MA, USA*, volume 218 of *LIPIcs*, pages 3:1–3:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. `doi:10.4230/LIPIcs.FORC.2022.3`.

3   Tal Alon, Magdalen Dobson, Ariel Procaccia, Inbal Talgam-Cohen, and Jamie Tucker-Foltz. Multiagent evaluation mechanisms. *In Proceedings of the AAAI Conference on Artificial Intelligence*, 34(02):1774–1781, April 2020. `doi:10.1609/aaai.v34i02.5543`.

4   Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. Steering user behavior with badges. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, pages 95–106, New York, NY, USA, 2013. Association for Computing Machinery. `doi:10.1145/2488388.2488398`.

5   Moshe Babaioff, Shahar Dobzinski, Sigal Oren, and Aviv Zohar. On bitcoin and red balloons. In Boi Faltings, Kevin Leyton-Brown, and Panos Ipeirotis, editors, *Proceedings of the 13th ACM Conference on Electronic Commerce, EC 2012, Valencia, Spain, June 4-8, 2012*, pages 56–73. ACM, 2012. `doi:10.1145/2229012.2229022`.

6   Maria-Florina Balcan, Dan DeBlasio, Travis Dick, Carl Kingsford, Tuomas Sandholm, and Ellen Vitercik. How much data is sufficient to learn high-performing algorithms? generalization guarantees for data-driven algorithm design. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, pages 919–932, New York, NY, USA, 2021. Association for Computing Machinery. `doi:10.1145/3406325.3451036`.

7   Yahav Bechavod, Katrina Ligett, Zhiwei Steven Wu, and Juba Ziani. Gaming helps! learning from strategic interactions in natural dynamics. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 1234–1242. PMLR, 2021. URL: `http://proceedings.mlr.press/v130/bechavod21a.html`.

8   Yahav Bechavod, Chara Podimata, Zhiwei Steven Wu, and Juba Ziani. Information discrepancy in strategic learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 1691–1715. PMLR, 2022. URL: `https://proceedings.mlr.press/v162/bechavod22a.html`.

9   Mark Braverman and Sumegha Garg. The role of randomness and noise in strategic classification. In Aaron Roth, editor, *1st Symposium on Foundations of Responsible Computing, FORC 2020, June 1-3, 2020, Harvard University, Cambridge, MA, USA (virtual conference)*, volume 156 of *LIPIcs*, pages 9:1–9:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. `doi:10.4230/LIPIcs.FORC.2020.9`.

10   Michael Brückner and Tobias Scheffer. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 547–555, New York, NY, USA, 2011. Association for Computing Machinery. `doi:10.1145/2020408.2020495`.

11   Moira Burke, Cameron Marlow, and Thomas M. Lento. Feed me: motivating newcomer contribution in social network sites. In Dan R. Olsen Jr., Richard B. Arthur, Ken Hinckley, Meredith Ringel Morris, Scott E. Hudson, and Saul Greenberg, editors, *Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, Boston, MA, USA, April 4-9, 2009*, pages 945–954. ACM, 2009. `doi:10.1145/1518701.1518847`.

12   Moira Burke and Burr Settles. Plugged in to the community: social motivators in online goal-setting groups. In Marcus Foth, Jesper Kjeldskov, and Jeni Paay, editors, *Proceedings of the Fifth International Conference on Communities and Technologies, C&T 2011, Brisbane, QLD, Australia, June 29 - July 2, 2011*, pages 1–10. ACM, 2011. `doi:10.1145/2103354.2103356`.

13   Emily Diana, Travis Dick, Hadi Elzayn, Michael Kearns, Aaron Roth, Zachary Schutzman, Saeed Sharifi-Malvajerdi, and Juba Ziani. Algorithms and learning for fair portfolio design. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, EC '21, pages 371–389, New York, NY, USA, 2021. Association for Computing Machinery. `doi:10.1145/3465456.3467646`.

14   Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, EC '18, pages 55–70, New York, NY, USA, 2018. Association for Computing Machinery. `doi:10.1145/3219166.3219193`.

15   David Easley and Arpita Ghosh. Incentives, gamification, and game theory: An economic approach to badge design. In *Proceedings of the Fourteenth ACM Conference on Electronic Commerce*, EC '13, pages 359–376, New York, NY, USA, 2013. Association for Computing Machinery. `doi:10.1145/2492002.2482571`.

16   Alex Frankel and Navin Kartik. Improving Information from Manipulable Data. *Journal of the European Economic Association*, 20(1):79–115, June 2021. `doi:10.1093/jeea/jvab017`.

17   Nika Haghtalab, Nicole Immorlica, Brendan Lucier, and Jack Z. Wang. Maximizing welfare with incentive-aware evaluation mechanisms. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 160–166. International Joint Conferences on Artificial Intelligence Organization, July 2020. Main track. `doi:10.24963/ijcai.2020/23`.

18   Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, ITCS '16, pages 111–122, New York, NY, USA, 2016. Association for Computing Machinery. `doi:10.1145/2840728.2840730`.

19   Keegan Harris, Hoda Heidari, and Zhiwei Steven Wu. Stateful strategic regression. *CoRR*, abs/2106.03827, 2021. `arXiv:2106.03827`.

20   Lily Hu, Nicole Immorlica, and Jennifer Wortman Vaughan. The disparate effects of strategic manipulation. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, pages 259–268, New York, NY, USA, 2019. ACM. `doi:10.1145/3287560.3287597`.

21   Jon Kleinberg and Manish Raghavan. How do classifiers induce agents to invest effort strategically? In *Proceedings of the 2019 ACM Conference on Economics and Computation*, EC '19, pages 825–844, New York, NY, USA, 2019. Association for Computing Machinery. `doi:10.1145/3328526.3329584`.

22   John Miller, Smitha Milli, and Moritz Hardt. Strategic classification is causal modeling in disguise. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6917–6926. PMLR, 2020. URL: `http://proceedings.mlr.press/v119/miller20b.html`.

23   Smitha Milli, John Miller, Anca D. Dragan, and Moritz Hardt. The social cost of strategic classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, pages 230–239, New York, NY, USA, 2019. Association for Computing Machinery. `doi:10.1145/3287560.3287576`.

24   D. Pollard. *Convergence of Stochastic Processes*. Springer New York, 1984. URL: `https://books.google.com/books?id=B2vgGMa9vd4C`.

25   Yonadav Shavit, Benjamin Edelman, and Brian Axelrod. Learning from strategic agents: Accuracy, improvement, and causality. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume abs/2002.10066 of *Proceedings of Machine Learning Research*, pages 8676–8686. PMLR, 13–18 July 2020. URL: `http://proceedings.mlr.press/v119/shavit20a.html`.

26   Ravi Sundaram, Anil Vullikanti, Haifeng Xu, and Fan Yao. Pac-learning for strategic classification. In *International Conference on Machine Learning*, pages 9978–9988. PMLR, 2021.

27   David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011.

28   Shenke Xiao, Zihe Wang, Mengjing Chen, Pingzhong Tang, and Xiwang Yang. Optimal common contract with heterogeneous agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7309–7316, April 2020. `doi:10.1609/aaai.v34i05.6224`.

A   Missing Proofs of Theorem 3

A.1   Proof of Theorem 4

▶ **Theorem 4.** *Algorithm 1 finds a set of targets that achieves the optimal social welfare (maximum total improvement) that is feasible using at most k targets given n agents. The algorithm runs in $\mathcal{O}(n^3)$.*

Proof. Proof of correctness follows by induction. Suppose that the value computed for all $T(\tau', \kappa')$ where $(\tau', \kappa') < (\tau, \kappa)$ is correct. Here "$<$" means (τ', κ') is computed before (τ, κ) which is when $\kappa' < \kappa$ and $\tau' \geq \tau$. First, if either $\tau = \tau_{\max}$ or $\kappa = 0$, the induction hypothesis holds since $T(\tau_{\max}, \kappa) = 0$ for all $1 \leq \kappa \leq k$, and $T(\tau, 0) = 0$, for all $\tau \in \mathcal{T}_p$. To show the inductive step holds note that the algorithm considers the optimal value for $T(\tau, \kappa)$ as the maximum of the $T(\tau', \kappa - 1) + \sum_{\tau \leq p_i < \tau' \text{ s.t. } \tau' - p_i \leq \Delta_i} (\tau' - p_i)$ over all the possible placement of the leftmost target τ'. Since $T(\tau', \kappa - 1)$ is computed correctly by the induction hypothesis and all the possible placements of the leftmost target are considered, the value obtained at $T(\tau, \kappa)$ is optimal and correct.

Now we proceed to bounding the time-complexity. There are $\mathcal{O}(nk)$ subproblems to be computed. Consider a pre-computation stage where $\sum_{\tau \leq p_i < \tau' \text{ s.t. } \tau' - p_i \leq \Delta_i} (\tau' - p_i)$ is computed for all pairs of $\tau, \tau' \in \mathcal{T}_p$. This stage takes $\mathcal{O}(n^3)$ time. Computation of each subproblem $T(\tau, \kappa)$ for all $\tau \in \mathcal{T}_p$ and $1 \leq \kappa \leq k$ requires $\mathcal{O}(n)$ operations. This is because to compute max in property 3), we compute $T(\tau', \kappa - 1) + \sum_{\tau \leq p_i < \tau' \text{ s.t. } \tau' - p_i \leq \Delta_i} (\tau' - p_i)$ for $\mathcal{O}(n)$ potential target levels greater than τ, for which each takes $\mathcal{O}(1)$ time. Since there are $\mathcal{O}(nk)$ subproblems, the running time of the algorithm is $\mathcal{O}(n^2 k + n^3) = \mathcal{O}(n^3)$.   ◀

B   Missing Proofs of Section 4

B.1   Proof of Theorem 5

▶ **Theorem 5.** *Algorithm 2 constructs the Pareto frontier for groups' social welfare using at most k targets given n agents in g groups, and has a running time of $\mathcal{O}(n^{g+2} k g \Delta_{\max}^g)$, where $\Delta_{\max}$ is the maximum improvement capacity.*

Proof. Proof of correctness follows by induction and it is along the same lines as proof of Algorithm 1. Suppose that Pareto-frontiers constructed for all $T(\tau', \kappa')$ where $(\tau', \kappa') < (\tau, \kappa)$ is correct. Here "$<$" means (τ', κ') is computed before (τ, κ) which is when $\kappa' < \kappa$ and $\tau' \geq \tau$. First, if either $\tau = \tau_{\max}$ or $\kappa = 0$, the induction hypothesis holds since $T(\tau_{\max}, \kappa) = \emptyset$ for all $1 \leq \kappa \leq k$, and $T(\tau, 0) = \emptyset$, for all $\tau \in \mathcal{T}_p$. The inductive step holds since the algorithm considers all the possible placement of the leftmost target τ'. Since $T(\tau', \kappa - 1)$ is computed correctly by the induction hypothesis and all the possible placements of the leftmost target are considered, the Pareto-frontier constructed at $T(\tau, \kappa)$ is correct.

Now we proceed to bounding the time complexity. Initially, in a pre-computation stage, for each pair of targets $\tau, \tau' \in \mathcal{T}_p$, $\sum_{\tau \leq p_i < \tau' \text{ s.t. } \tau' - p_i \leq \Delta_\ell} \mathbb{1}\{i \in G_\ell\}(\tau' - p_i)$ is pre-computed for all groups and is stored in a tuple of size g. This stage can be done in $\mathcal{O}(n^3)$. Each set $T(\tau, \kappa)$ has size at most $(n\Delta_{\max} + 1)^g$, since each individual can move for one of the values $\{0, \cdots, \Delta_{\max}\}$ and therefore, the total improvement in each group is one of the values $\{0, \cdots, n\Delta_{\max}\}$. At each step of the recurrence, given the information stored in the pre-computation stage, the summation can be computed in $\mathcal{O}(g)$. When computing a subproblem $T(\tau, \kappa)$, the recurrence searches over $\mathcal{O}(n)$ targets $\tau' \in \mathcal{T}_p$, and at most $(n\Delta_{\max} + 1)^g$ tuples of group improvement in $T(\tau', \kappa - 1)$. As a result, solving each subproblem takes $\mathcal{O}(ng(n\Delta_{\max})^g)$. The total number of subproblems that need to get solved is $\mathcal{O}(nk)$. Therefore, the total running time of the algorithm is $\mathcal{O}(n^{g+2} k g \Delta_{\max}^g + n^3) = \mathcal{O}(n^{g+2} k g \Delta_{\max}^g)$.   ◀

B.2   Proof of Corollary 6

▶ **Corollary 6.** *There is an efficient algorithm that finds a set of at most k targets that maximizes minimum improvement across all groups, i.e., maximizing $\min_{1 \le \ell \le g} SW_\ell$.*

Proof. Algorithm 2 constructs the Pareto frontier for groups' social welfare. By iterating through all Pareto-optimal solutions, we can find the solution that maximizes the minimum improvement across all groups. There are at most $(n\Delta_{\max} + 1)^g$ Pareto-optimal solutions. Finding the minimum improvement in each solution takes $\mathcal{O}(g)$. Therefore, in total, finding the solution that maximizes the minimum improvement across all groups takes $\mathcal{O}(g(n\Delta_{\max})^g)$.   ◀

C   An FPTAS for Maximizing Minimum Group Improvement

In this section, we present a Fully Polynomial Time Approximation Scheme (FPTAS) to maximize minimum improvement across all groups. Here, we assume that each group ℓ has its own improvement capacity Δ_ℓ. The algorithm finds a set of at most k targets that approximates the max-min objective within a factor of $1 - \varepsilon$ for any arbitrary value of $\varepsilon > 0$. Here, we relax the assumption on the integrality of p_i, Δ_i values needed for the running time guarantee in Theorem 5, and suppose all p_i, Δ_i values are real numbers. Similar to the dynamic program based on Algorithm 2, for each subproblem, a set containing all g-tuples of improvements $(I_1, I_2, \cdots, I_g)$ that are simultaneously achievable for all groups is stored. However, computing all such tuples takes exponential time since $\sum_{i=1}^{k} \binom{2n}{i}$ possible cases of targets' placements need to be considered. Therefore, we discretize the set of all possible improvements for this problem by rounding all the improvement tuples, and develop an FPTAS algorithm. The recurrence for the dynamic program is given in Algorithm 4. The algorithm runs efficiently when the number of groups is a constant.

▶ **Algorithm 4.** *The algorithm considers two separate cases of $k < g$, and $k \ge g$. For the $k \ge g$ case, the algorithm finds a set of k targets that approximates the max-min objective within a factor of $1 - \varepsilon$ for any arbitrary value of $\varepsilon > 0$. For the $k < g$ case, it finds an optimal solution for the max-min objective.*

For the $k \ge g$ case, there exists an FPTAS for the max-min objective as follows. First, run a dynamic program using the following recursive function to get a set of Pareto-optimal solutions. In this Pareto-frontier, we show the solution that maximizes minimum improvement across all groups, gives a $(1 - \varepsilon)$-approximation for the max-min objective. In the recurrence, $\mu_\ell = \varepsilon \Delta_\ell / (16 k g^3)$ for $1 \le \ell \le g$, and Δ_ℓ is the improvement capacity of agents in group ℓ.

$$\mathcal{F}(\tau', k') = \left\{ \left(\mu_\ell \left\lfloor \frac{I'_\ell + \left(\sum_{\substack{\tau' \le p_i < \tau \\ s.t. \ \tau - p_i \le \Delta_\ell}} \mathbb{1}\{i \in G_\ell\}(\tau - p_i) \right)}{\mu_\ell} \right\rfloor \right)_{\ell=1}^{g} , \right.$$

$$\left. s.t. \ (I'_\ell)_{\ell=1}^{g} \in \mathcal{F}(\tau, k' - 1), \tau \in \mathcal{T}_p, \tau \ge \tau' \right\}$$

Intuitively, $\mathcal{F}(\tau', k')$ stores the rounded down values of the feasible tuples of group improvements when all agents on or to the right of τ' are available and k' targets are used. The corresponding set of targets used to construct the improvement tuples in $\mathcal{F}(\tau', k')$ is kept in a hash table $\mathcal{S}(\tau', k')$, whose keys are the improvement tuples in $\mathcal{F}(\tau', k')$. The dynamic program ends after computing $\mathcal{F}(\tau_{\min}, k)$ and $\mathcal{S}(\tau_{\min}, k)$. At the end, we output the set of targets in $\mathcal{S}(\tau_{\min}, k)$ corresponding to the improvement tuple that maximizes the improvement of the worst-off group. Lemma 27 shows that this algorithm gives a $(1 - \varepsilon)$-approximation for the max-min objective when $k \ge g$.

When $k < g$, for each subset of $\mathcal{T}_p$ of size at most k that is corresponding to a placement of targets, we store its corresponding improvement tuple. Next, we iterate through all improvement tuples and output the one that maximizes minimum improvement.

▶ **Lemma 27.** *Algorithm 4 gives a $(1 - \varepsilon)$-approximation for the max-min objective when $k \geq g$.*

Proof. The proof is by induction. Consider an improvement tuple $(I_1, \cdots, I_g)$ corresponding to an arbitrary set of $k - 1$ targets, and let $(I'_1, \cdots, I'_g)$ denote the rounded down values where $I'_\ell = \mu_\ell \lfloor \frac{I_\ell}{\mu_\ell} \rfloor$ for all $1 \leq \ell \leq g$. Suppose that for all $1 \leq \ell \leq g$, $I'_\ell \geq I_\ell - (k-1)\mu_\ell$.

Now consider an improvement tuple $(J_1, \cdots, J_g)$ corresponding to an arbitrary set of k targets. For each $1 \leq \ell \leq g$, let $J'_\ell = \mu_\ell \lfloor \frac{J_\ell}{\mu_\ell} \rfloor$. We show that for each $1 \leq \ell \leq g$, $J'_\ell \geq J_\ell - k\mu_\ell$. For all $1 \leq \ell \leq g$, let $J_\ell = L_\ell + I_\ell$, where L_ℓ is the improvement of group ℓ that the leftmost target provides, and I_ℓ captures the true improvement of group ℓ that the remaining $k - 1$ targets provide. Let $I'_\ell = \mu_\ell \lfloor \frac{I_\ell}{\mu_\ell} \rfloor$. Then $J'_\ell = \mu_\ell \lfloor \frac{L_\ell + I'_\ell}{\mu_\ell} \rfloor$ implying that $J'_\ell \geq L_\ell + I'_\ell - \mu_\ell$. By the induction hypothesis, $I'_\ell \geq I_\ell - (k-1)\mu_\ell$. Therefore,

$$J'_\ell \geq L_\ell + I'_\ell - \mu_\ell \geq L_\ell + I_\ell - (k-1)\mu_\ell - \mu_\ell = L_\ell + I_\ell - k\mu_\ell$$

Therefore, for each set of k targets, the rounded improvement of each group ℓ stored in the table is within an additive factor of $k\mu_\ell = \varepsilon\Delta_\ell/(16g^3)$ of its true improvement. We argue that in the solution returned by the algorithm, improvement of each group is at least $(1 - \varepsilon)OPT$. First, when $k \geq 1$, each group can improve for at least Δ_ℓ by setting a target within a distance of Δ_ℓ from its rightmost agent. Now, using Theorem 7 when $k \geq g$, there exists a solution that is simultaneously $1/(16g^3)$-optimal for all groups. Therefore, the optimum value of the max-min objective is at least $OPT \geq \Delta_\ell/(16g^3)$ for all $1 \leq \ell \leq g$. Therefore, for each solution consisting of k targets, the rounded improvement of each group is within an additive factor of εOPT of its true improvement. As a result, the minimum group improvement in the returned solution is at least $(1 - \varepsilon)OPT$.   ◀

In the following, we bound the approximation factor of our algorithm in both cases of $k \geq g$ and $k < g$.

▶ **Corollary 28.** *Algorithm 4 described above gives a $(1 - \varepsilon)$-approximation for the max-min objective.*

Proof. For the case of $k \geq g$, by Lemma 27 the algorithm outputs a $(1 - \varepsilon)$-approximation. For $k < g$, it outputs an optimum solution. Therefore, in total, it gives a $(1-\varepsilon)$-approximation for the max-min objective.   ◀

In the following, we bound the time-complexity of the algorithm.

▶ **Theorem 29.** *Algorithm 4 has a running time of $\mathcal{O}(n^{g+2}k^{g+1}g^{3g+1}/\varepsilon^g)$.*

Proof. Initially, in a pre-computation stage, for each pair of targets $\tau, \tau' \in \mathcal{T}_p$, $\sum_{\tau' \leq p_i < \tau \text{ s.t. } \tau - p_i \leq \Delta_\ell} \mathbb{1}\{i \in G_\ell\}(\tau - p_i)$ is pre-computed for all groups and is stored in a tuple of size g. This stage can be done in $\mathcal{O}(n^3)$. Now, first consider the case where $k \geq g$. We show the dynamic programming algorithm using recurrence $\mathcal{F}(\tau', k')$ has a running time of $\mathcal{O}(n^{g+2}k^{g+1}g^{3g+1}/\varepsilon^g)$. Each set $\mathcal{F}(\tau', k')$ and $\mathcal{S}(\tau', k')$ has size at most $\prod_{\ell=1}^{g}(n\Delta_\ell/\mu_\ell)^g = (16nkg^3/\varepsilon)^g$. At each step of the recurrence, given the information stored in the pre-computation stage, the summation can be computed in $\mathcal{O}(g)$. When computing $\mathcal{F}(\tau', k')$, the recurrence searches over $\mathcal{O}(n)$ targets $\tau \in \mathcal{T}_p$, and at most

$\prod_{\ell=1}^{g}(n\Delta_\ell/\mu_\ell)^g = (16nkg^3/\varepsilon)^g$ tuples of group improvement in $\mathcal{F}(\tau, k'-1)$. As a result, solving each subproblem takes $\mathcal{O}(ng(nkg^3/\varepsilon)^g)$. The total number of subproblems that need to get solved is $\mathcal{O}(nk)$. Therefore, the total running time of computing $\mathcal{F}(\tau_{\min}, k)$ is $\mathcal{O}(n^{g+2}k^{g+1}g^{3g+1}/\varepsilon^g)$.

Next, consider the case where $k < g$. The algorithm considers $\mathcal{O}(n^g)$ placements of targets. Given the pre-computation stage, computing the improvement tuple corresponding to each placement of targets takes $\mathcal{O}(kg)$. As a result, this case takes $\mathcal{O}(kgn^g)$.

Therefore, the total running time of algorithm is $\mathcal{O}(n^3 + n^{g+2}k^{g+1}g^{3g+1}/\varepsilon^g + kgn^g) = \mathcal{O}(n^{g+2}k^{g+1}g^{3g+1}/\varepsilon^g)$. ◀

D   Distance between consecutive target levels

Observation 13 shows it is without loss of optimality to assume the distance between every other targets is at least Δ in the common improvement capacity model. The following example investigates this property for *consecutive* targets, and shows an instance where the distance between two consecutive targets is arbitrarily small compared to Δ in the optimal solution.

▶ **Example 30.** Suppose $\Delta = 1$ and there is no limit on the number of targets. Suppose there is an agent at position 0, an agent at position 1, and m agents at position $1 + 1/m$. The optimal solution is $\mathcal{T} = \{\tau_1 = 1, \tau_2 = 1 + 1/m, \tau_3 = 2 + 1/m\}$. As $m \to \infty$, the distance between τ_1 and τ_2 approaches 0.

E   Extensions and Open Problems

E.1   Extension 1: A lower bound on the number of agents that improve

Consider Algorithm 1 whose goal is to find a set of at most k target levels that maximizes the total improvement for a collection of n agents. It is possible that the solution of this algorithm focuses on a small fraction of the agents and does not help many agents to improve. In Algorithm 5, we show how to modify Algorithm 1 to ensure at least $n_{\ell b}$ agents improve. The main idea for the recursive step (item 4 in Algorithm 5) is to first consider the potential leftmost targets $\tau' > \tau$, let x denote the number of agents that are within reach to τ', and use the smaller subproblem of finding the optimal targets for agents on or to the right of τ' with one less available target level and an updated lower bound of $\eta - x$, i.e., $S(\tau', \kappa - 1, \eta - x)$. We add the performance of each potential leftmost target to the optimal improvement of the remaining subproblem and pick the leftmost target that maximizes this summation.

▶ **Algorithm 5.** *Run dynamic program based on function S, defined below, that takes $\cup_i\{p_i\}$ and k as input and outputs $S(\tau_{\min}, k, n_{\ell b})$, as the optimal improvement, and $S'(\tau_{\min}, k, n_{\ell b})$, as the optimal set of targets; where $\tau_{\min} = \min\{\tau \in \mathcal{T}_p\}$ and $\tau_{\max} = \max\{\tau \in \mathcal{T}_p\}$. $S(\tau, \kappa, \eta)$ captures the maximum improvement possible for agents on or to the right of $\tau \in \mathcal{T}_p$ when κ target levels can be selected and at least η agents need to improve. If $S(\tau_{\min}, k, n_{\ell b}) = -\infty$ then incentivizing at least n_{lb} agents to improve is impossible. Function S is defined as follows.*

1) *For any $\tau \in \mathcal{T}_p, \eta \geq 1$, we have $S(\tau, 0, \eta) = -\infty$.*

2) *For any $1 \leq \kappa \leq k, \eta \geq 1$, $S(\tau_{max}, \kappa, \eta) = -\infty$, where $\tau_{max} = \max\{\tau \in \mathcal{T}_p\}$. This holds since no agents can improve to τ_{max}, however at least η agents to the right of τ_{max} need to improve which is a contradiction.*

3) *For any $\tau \in \mathcal{T}_p, 0 \leq \kappa \leq k, \eta \leq 0$, $S(\tau, \kappa, \eta) = T(\tau, \kappa)$ where function T is defined in Algorithm 1.*

4) *For any $\tau \in \mathcal{T}_p, \tau < \tau_{max}$, $1 \leq \kappa \leq k$, and $1 \leq \eta \leq n$:*

$$S(\tau, \kappa, \eta) = \max_{\substack{\tau' \in \mathcal{T}_p \\ s.t \ \tau' > \tau}} \left(S(\tau', \kappa - 1, \eta - \mathbb{1}\left[i \mid \tau \leq p_i < \tau' \ s.t. \ \tau' - p_i \leq \Delta_i \right]) + \sum_{\substack{\tau \leq p_i < \tau' \\ s.t. \ \tau' - p_i \leq \Delta_i}} (\tau' - p_i) \right)$$

$S'(\tau, \kappa, \eta)$ *keeps track of the optimal set of targets corresponding to* $S(\tau, \kappa, \eta)$.

E.2   Extension 2: Optimizing the number of target levels

The nonmonotonicity property may make adding a new target level to the current placement reduce the maximum improvement (see Figure 1b), or wasteful if we place the new target level somewhere no agent can reach or on top of an existing target. Therefore, when considering $k = 1, 2, \ldots, n$, it is possible that the maximum total improvement is achieved at $k < n$. Using the dynamic program based on Algorithm 1 we can find the minimum value of k that satisfies this property and minimizes the number of targets subject to achieving maximum total improvement. Furthermore, by finding the total amount of improvement for different values of k, the principal can decide how many targets are sufficient to achieve a desirable total improvement (bi-criteria objective).

E.3   Open Problem: Tightening the approximation gap

Algorithm 3, as stated in Theorem 7, provides an $\Omega(1/g^3)$-approximation simultaneous guarantee compared to the optimal solution for each group using at most k targets; and as stated in Theorem 20, provides an $\Omega(1/g^2)$-approximation simultaneous guarantee compared to the optimal solution for each group using at most $\lceil k/g \rceil$ targets. Example 9, on the other hand, shows an instance where no solutions with $> 1/g$ simultaneous approximation for the groups is possible for either of the benchmarks. Therefore, there is a gap of $\mathcal{O}(g^2)$ for the first, and a gap of $\mathcal{O}(g)$ for the second benchmark. Finding the optimal order of approximation guarantees for these benchmarks and tight lower bounds are the main problems left open by our work.