Regret Analysis of Repeated Delegated Choice

MohammadTaghi Hajiaghayi, Mohammad Mahdavi, Keivan Rezaei, Suho Shin

University of Maryland, College Park
{hajiagha,mahdavi,krezaei,suhoshin}@umd.edu

Abstract

We present a study on a repeated delegated choice problem, which is the first to consider an online learning variant of Kleinberg and Kleinberg, EC'18. In this model, a principal interacts repeatedly with an agent who possesses an exogenous set of solutions to search for efficient ones. Each solution can yield varying utility for both the principal and the agent, and the agent may propose a solution to maximize its own utility in a selfish manner. To mitigate this behavior, the principal announces an eligible set which screens out a certain set of solutions. The principal, however, does not have any information on the distribution of solutions nor the number of solutions in advance. Therefore, the principal dynamically announces various eligible sets to efficiently learn the distribution. The principal's objective is to minimize cumulative regret compared to the optimal eligible set in hindsight. We explore two dimensions of the problem setup, whether the agent behaves myopically or strategizes across the rounds, and whether the solutions yield deterministic or stochastic utility. We obtain sublinear regret upper bounds in various regimes, and derive corresponding lower bounds which implies the tightness of the results. Overall, we bridge a well-known problem in economics to the evolving area of online learning, and present a comprehensive study in this problem.

1 Introduction

Delegation is perhaps one of the most frequent economic interactions one may see around in real life (Holmstrom 1980; Bendor, Glazer, and Hammond 2001; Amador and Bagwell 2013). Abstractly speaking, consider a principal with less information who tries to find an optimal solution from an agent with expertise, but there's an information asymmetry such that she[1] *cannot directly access* the solutions that the agent possesses (Alonso and Matouschek 2008; Kleinberg and Kleinberg 2018; Kleiner 2022; Hajiaghayi, Rezaei, and Shin 2023). Instead, she requires the agent to propose a set of solutions and then commits to the final one among them. The principal and the agent, however, may have *misaligned utility* for the solution selected, and thus the agent may propose a solution in a selfish manner. To cope with it, the principal announces a set of eligible solutions before the agent proposes, and only accepts the eligible solution.

To provide a concrete example, consider (online) labor market or crowdsourcing platform such as Upwork. We have a task requester (principal) who regularly visits the platform (agent) and tries to solve a series of tasks. The platform has a pool of workers (solutions). At each time the requester visits, the platform recommends some set of workers, and the requester selects a single worker to commit to the task. Obviously, the task requester wants to hire a qualified worker. The platform, on the other hand, aims to maximize its long-term revenue by recommending workers who solve tasks quickly, even if their quality is not high, allowing them to be assigned to other tasks promptly. This misalignment of utility may lead to the platform strategically recommending unqualified workers. To mitigate this, the requester sets restrictions, such as requiring certificates in specific areas like a foreign language or web development, when requesting worker recommendations. We refer to Appendix A in the full paper for more examples on motivations.

If the task requester is fully aware of the set of workers that the platform has, then she can directly impose a strong restriction to make the platform recommend the specific workers she wants. In practice, however, such information is not feasible priorly, instead, the requester needs to *learn the distribution* of existing workers in the repeated interaction. The fundamental question here is, how the requester should dynamically determine which sort of restriction to impart at each round, in order to maximize cumulative utility over the set of tasks. Furthermore, one may ask what happens if the platform also tries to *strategize across the rounds* to deceive the requester, and what if the quality of each worker is not fixed in advance, but rather is given from a *latent distribution*.

This work introduces the *repeated delegated choice* problem, which focuses on how the principal can design an efficient delegation mechanism. To the best of our knowledge, this is the first study to explore an online learning extension of the delegated choice problem presented by (Armstrong and Vickers 2010; Kleinberg and Kleinberg 2018; Hajiaghayi, Rezaei, and Shin 2023). In our model, the principal lacks initial information about the solutions' distribution. Instead, through repeated announcements of eligible sets that may screen out some solutions, the principal aims to learn the solutions' distributions in a sample-efficient manner. The

[1] Feminine pronouns (masculine) hereafter denote the principal (agent).

Utility \ Behavior	Myopic	Strategic
Deterministic	Theorem 3.1	Theorem 3.4,3.8
Stochastic	Theorem 4.1	Theorem 4.3

Table 1: Summary of our results under different settings.

principal aims to minimize cumulative regret compared to the optimal eligible set in hindsight.

We distill the problem into two dimensions of whether the utility of each solution is deterministic or stochastic, and whether the agent strategizes across the rounds or not, and provide a comprehensive regret analysis for each setting. In the myopic agent setting, the agent plays a best-response to the eligible set at each round, *i.e.,* a strategy which maximizes *myopic utility* without regard to the future utility and resulting behavior of the mechanism.[2] Hence, the principal's objective boils down to efficiently learning the distribution of utilities by selecting proper eligible sets at each round, while only observing the partial feedback from the choice of eligible set, *i.e.,* which solution the agent submits (or possibly declines to submit any). This challenge intensifies with a strategic agent, as the agent may intentionally hide solutions or deviate from their best response for greater utility in later rounds. Consequently, the feedback is not guaranteed to be stochastic across rounds, making the analysis more complex.

Our contributions. We here provide a summary of our contributions and techniques. All the proofs can be found in the appendix in the full paper. The results are summarized in Table 1. First, we observe a revelation-principle-style-of-result such that it suffices to focus on a class of so-called single-proposal mechanism, formally defined in Definition 2.1. Interestingly, we show that the myopic deterministic setting can be reduced to the repeated posted price mechanism (RPPM) with myopic buyer. [3] Denoting the principal's utility for each solution by X_i, one can effectively construct an instance of RPPM by converting X_i to the buyer's value $v = \max_i X_i$ in RPPM. In both problems, the optimal benchmark is to obtain $\max_i X_i$, and the reduction follows. Combined further with an iterative algorithm, we obtain a regret upper bound of $O(\min(K, \log\log T))$, where K denotes the number of solutions and T is the time horizon.

With stochastic valuation, however, this does not work since the benchmark in RPPM is to put an ex-ante best fixed price, which does not coincide $\max_i X_i$. Indeed, we observe that the optimal benchmarks cannot be reduced from one to another in general. Instead, we mainly reduce our problem to a stochastic multi-armed bandit problem via proper discretization over the space of eligible sets equipped with a variant of analysis by (Kleinberg and Leighton 2003), and obtain a regret of $O(\sqrt{T\log T})$ under the same assumption imposed in (Kleinberg and Leighton 2003).

For the strategic agent with deterministic utility, we first observe that it is necessary to impose a certain assumption on the agent's utility sequence to obtain positive results. Precisely, the agent with non-discounting utility can strategize so that no algorithm can obtain sublinear regret, where the formal proof is presented in Appendix K. In this context, to capture both of the practicality and theoretical tractability, we consider γ-discounting strategic agent whose utility is discounted by a multiplicative factor of γ at each round. We also note that this is common in the literature (Amin, Rostamizadeh, and Syed 2013; Haghtalab et al. 2022).

Given that, for the γ-discounting agent with deterministic utility, we first consider a case in which the agent's utility is uniformly bounded below by $y_{\min}$ and the principal is aware of it. In this setting, by exploiting the delay technique of (Haghtalab et al. 2022), we obtain a regret bound of $O(KT_\gamma \log \frac{T_\gamma}{y_{\min}})$ where $T_\gamma = 1/(1-\gamma)$. The dependence on K can be replaced by $\log T$ by shrinking the eligible set more in an aggressive manner, thereby obtaining a regret of $O(T_\gamma \log \frac{T_\gamma}{y_{\min}} + \log T)$. Note that these bounds yield sublinear regret only if $y_{\min} = e^{-o(T)}$. We complement these results by showing that any algorithm suffers regret of $\Omega(T)$ if $y_{\min} \le e^{-T}$.

On the other hand, if the agent's minimum utility is not known or unbounded, there's no guarantee that the agent behaves myopically for any delay that is imposed in the algorithm. Instead, under minor assumptions that the solutions are densely spread with respect to parameter d and Lipschitzness between the principal's and agent's utilities, we obtain an efficient algorithm that achieves a regret upper bound of $O(T_\gamma \log \frac{T_\gamma}{\alpha} + \log \frac{1}{d} + dT)$, where α is a function of the Lipschitz parameters. The linear dependence of $O(dT)$ regret may look a lot at first glance, we observe this is inevitable for any algorithm, thereby justifying our assumption.

In the stochastic setting with γ-discounting strategic agent, we reuse the machinery by (Haghtalab et al. 2022; Lancewicki et al. 2021), and obtain a regret of $O(\sqrt{T\log T})$. More specifically, we can view the proposed solution as a perturbed output of a stochastic bandit, where the perturbation comes from the agent's strategic behavior. The technical subtlety lies on how we should upper/lower bound such perturbed output to properly apply (Lancewicki et al. 2021), i.e., how we should construct a random perturbation interval.

Related Works

Delegation. Dating back to the seminal work of (Holmstrom 1980), a number of literature from the economics community study the theory of delegation, mostly within the extent of characterizing the regimes under which some simple mechanisms reach the optimal solution (Alonso and Matouschek 2008; Armstrong and Vickers 2010; Kleiner 2022). Recently, (Kleinberg and Kleinberg 2018) study a problem of *dele-*

[2] This model of myopic agent accommodates a perspective of "multiple agents" setting in which at each round an agent having the same type of solutions arrive and interacts with the principal. In this viewpoint, the agents are bound to be myopic due to the single round interaction per agent.

[3] Overall, we observe an intimate connection between the RPPM and our problem under certain settings. In general, however, our problem spawns additional challenges of having multiple latent random variables and the principal is even unaware of the number of potential solutions. We provide more detailed discussion in Appendix B in the full paper.

gated choice[4] with a lens of computer science, and show that there exists a mechanism with 2-approximation compared to the case in which the principal can fully access all the solutions in advance, based on a novel connection to prophet inequality problem (Samuel-Cahn 1984). Their result, however, depends on the assumption that the principal knows the distribution from which the utility of each solution is drawn, *i.e.,* they study the efficiency of Bayesian mechanism. Interestingly, if the principal has no such information at all, *i.e.,* prior-independent mechanism, the result becomes largely pessimistic, *i.e.,* there exists a problem instance in which the principal's approximation becomes arbitrarily bad. (Hajiaghayi, Rezaei, and Shin 2023) reveal that prior-independent mechanisms can be made efficient with multiple agents, but this does not hold with a single agent.

Repeated delegation. (Lipnowski and Ramos 2020) study a problem of infinitely repeated delegation, however, their model of delegated choice is largely different from ours. Mainly, their model considers aligned utility but when the principal bears the cost of adopting a project. Their objective is to persuade the agent to adopt the project when it is truly good, whereas the agent tries to always adopt the project. Several lines of work (Li, Matouschek, and Powell 2017; Guo and Hörner 2021) study a repeated game of project choice, but we do not discuss it in details due to significant differences from our model. A line of work (Lewis 2012; Xiao, Hu, and Wang 2022) study a delegated search problem, especially a dynamic version by (Rahmani and Ramachandran 2016), but the players bear the cost of search for solutions in their model, whereas the solutions are exogenous to the mechanism in our model.

Stackelberg games. Our problem can be viewed as an online learning version of repeated Stackelberg game (Von Stackelberg 2010; Marecki, Tesauro, and Segal 2012; Bai et al. 2021; Lauffer et al. 2022; Zhao et al. 2023). A common objective in this area of work is to minimize a Stackelberg regret, *i.e.,* difference to the optimal policy that knows the leader's optimal action in hindsight, and the above works aim to minimize the cumulative Stackelberg regret of a leader, assuming that a follower best responds at each round. Especially, our model of strategic agent belongs to the growing area of learning in games with strategic agent (Birmpas et al. 2020; Haghtalab et al. 2022; Zhao et al. 2023). More precisely, (Birmpas et al. 2020) study how the follower can efficiently deceive the leader by misreporting his valuation. (Haghtalab et al. 2022) proposes a generic delaying technique to deal with a strategic agent, and proposes several applications to strategic classification, repeated posted price mechanism (henceforth RPPM), and Stackelberg security game. Indeed, our model resembles RPPM of (Kleinberg and Leighton 2003; Amin, Rostamizadeh, and Syed 2013; Babaioff et al. 2017). However, RPPM restricts the buyer and the seller's utility to be linearly negatively correlated, but our model accommodates any kind of correlation. In addition, our agent has multiple solutions to choose from compared to only accept/reject of RPPM, and thus is technically more challenging to predict/analyze the agent's strategic behavior.

2 Problem Setup

In a *repeated delegated choice* problem, there is a principal and an agent. The agent is equipped with a set of solutions $A = \{a_0, a_1, \dots, a_K\}$ where K denotes the cardinality of the set of possible solutions[5], and a_0 denotes the null solution $\perp$ which means that the agent submits nothing. At each round $t \in [T]$, solution a incurs a nonnegative random utility for the principal and the agents. Denote by $X_a^{(t)}$ the utility random variable (r.v.) of the principal selecting the solution a and $Y_a^{(t)}$ the random utility of the agent given solution a, both of which has support in $\Omega := [0, 1]$. The random vector $(X_a^{(t)}, Y_a^{(t)})$ is independent and identically distributed (i.i.d.) for $t \in [T]$. Importantly, the agent can *access* the ex-post utility of all the solutions $a \in A$ at each round, but the principal cannot. The agent is equipped with a discounting factor $\gamma \in (0, 1)$, *i.e.,* he discounts the utility at round t by a factor of γ^{t-1}. That is, the agent's true utility for solution a at round t is $\gamma^{t-1} Y_a^{(t)}$. This assumption on agent regret is common in studies concerning strategic agents (Amin, Rostamizadeh, and Syed 2013; Haghtalab et al. 2022). It is shown in (Amin, Rostamizadeh, and Syed 2013) that in the repeated posted-price mechanism problem, a sublinear regret can not be achieved for a non-discounting strategic agent. This is also the case in our problem, as a non-discounting agent might have the incentive to hide a solution that is worse than another solution in terms of agent utility but better for the principal. We further explore this in Appendix K. Define $T_\gamma = 1/(1-\gamma)$. Given a mechanism M, at each round $t \in [T]$, the agent chooses a (possibly random) subset of solutions $S^{(t)} \subset A$, and submits them to the principal. We write 2^X to denote the power set of a set X, and $\Delta(X)$ for a simplex over X. Thus, the agent's action belongs to $S^{(t)} \in \Delta(2^A)$.

History, mechanism, and agent's policy. At each round $t \in [T]$, the mechanism determines which solutions to commit given the agent's action $S^{(t)}$. This choice is based on the history available up to round t, formally defined by $H_t := \cup_{l=1}^{t-1}(S^{(l)}, a^{(l)})$, where $a^{(l)}$ denotes the solution selected at round l. We define $\mathcal{H} := \cup_{t \geq 1}(2^A, A)$ to be the set of all possible histories of the game, *i.e.,* each H_t is a subset of $\mathcal{H}$. Formally, the mechanism $M : \mathcal{H} \times 2^A \mapsto A$ specifies which solutions to select at each round t given the history $H_t \in \mathcal{H}$ and the agent's submission. Importantly, the mechanism is only able to choose an action among the actually submitted solutions by the agent. Also, the principal *commits* to a mechanism before the game starts.

Let $\mathcal{M}$ be the set of all possible mechanisms. Correspondingly, the agent's policy $P : \mathcal{H} \times \mathcal{M} \mapsto \Delta(2^A)$ is a function that takes the mechanism announced by the principal and the

[4] They consider two types of problem settings, one of which is delegated search with sampling costs, and the other is delegated choice, referring back to (Armstrong and Vickers 2010). Since we also assume that the solutions of the agent are exogenous to mechanisms, we frame our model as a delegated choice problem.

[5] We do not restrict the number of solutions to be finite, or constant with respect to T.

history sequence H_t and decides an (possibly randomized) action. Let $\mathcal{P}$ be the set of all possible agent policies. We write s_t to denote the solution eventually selected at round t by the mechanism. Note that s_t can be null, *i.e.,* $s_t = \perp$, if the principal declines to accept any proposed solution. In this case, both the agent's and the principal's utilities are zero. We write $X^{(t)}_{M,P}$ and $Y^{(t)}_{M,P}$ to denote the principal's and agent's utility at round t under M and P. Define $\Phi_{M,P} = \mathbb{E}[X^{(t)}_{M,P}]$ and $\Psi_{M,P} = \mathbb{E}[Y^{(t)}_{M,P}]$ to denote the expected utility of the principal and the agent, respectively.

Mechanism description. Overall, the interaction between the principal and the agent proceeds as follows:

i The principal commits to a mechanism.

ii At each round, agent observes the realized solutions and their utility.

iii Agent (possibly strategically) proposes solutions.

iv Principal observes the proposed solutions and corresponding utility, and determines the final outcome with respect to the committed mechanism.

v Steps ii-iv are repeated.

Single-proposal mechanism. We mainly deal with the following specific type of mechanism, inspired by (Kleinberg and Kleinberg 2018).

Definition 2.1 (Single-proposal mechanism)**.** In a single proposal mechanism M, at each round t, the principal announces an eligible set $E^{(t)} \subset \Omega^2$, and the agent submits only a single solution a. If $(X^{(t)}_a, Y^{(t)}_a) \in E^{(t)}$, then the principal accepts the solution, otherwise, she selects nothing.

We further say that a mechanism is threshold-based, if its eligible set only puts a (possibly strict) lower bound on the principal's utility. We define $E_\tau = \{a : X_a \geq \tau\}$ and $E^>_\tau = \{a : X_a > \tau\}$ to represent threshold-based eligible sets for a threshold τ. Given a single proposal mechanism, we write $x^{(t)}_a$ and $y^{(t)}_a$ to denote the eventual utility of the principal and the agent at round t when the agent proposes solution a, *i.e.,* which reflects the principal's decision.

Notably, we provide a revelation principle style of result which states that any mechanism can be reduced to a single-proposal mechanism.

Theorem 2.2. *Given any mechanism M and any agent's policy P, there exists a single-proposal mechanism M' and corresponding deterministic agent's policy P' such that $\Phi_{M,P} \leq \Phi_{M',P'}$ and $\Psi_{M,P} \leq \Psi_{M',P'}$.*

Thanks to the reduction above, we can essentially focus on the single-proposal mechanism, and the agent only needs to determine which solution to submit at each round. Thus, unless specified explicitly, we now focus on the single-proposal mechanism. Note that the reduction from any deterministic mechanism with deterministic policy follows from a variant of the proof of the standard revelation principle (Nisan et al. 2007). For randomized policy, we can reduce it to a deterministic policy by sequentially derandomizing each round's random events in a backward manner.

Approximately best response and Stackelberg regret. Our construction of a mechanism against a strategic agent requires a notion of approximate best response of the agent, defined as follows.

Definition 2.3 (ε-best response)**.** Given a mechanism M and history H_t, let A_E be a union of the set of eligible solutions given eligible set E and the null outcome $\perp$. Then, the ε-best response at round t for eligible set E is defined by

$$\textsc{Br}^{(t)}_\varepsilon(E) = \{a \in A_E : y^{(t)}_a \geq y^{(t)}_{a'} - \varepsilon, \forall a' \in A\}.$$

If $\varepsilon = 0$, we simply say best response and denote by $\textsc{Br}^{(t)}$.

Whenever there are multiple solutions as best response, we assume that a myopic agent plays in favor of the principal, *i.e.,* submits the solution that maximizes the principal's utility.

Fundamentally, the dynamics of the single-proposal mechanism belongs to a repeated Stackelberg game in which the principal moves first by announcing an eligible set, and then the agent follows by proposing solutions, at each round. In repeated Stackelberg games (possibly with strategic agent), typical objective is to minimize a cumulative regret compared to the case when the mechanism knows the optimal eligible set in hindsight, and the agent *myopically* responds to the principal's move. In our setting, this benchmark boils down to the case under which the mechanism knows the distribution of $X^{(t)}_a$ and $Y^{(t)}_a$ in hindsight, while the agent best responds to the principal's eligible set at each round. In this case, the optimal principal's utility can be written as,

$$\textsc{Opt} = \max_{E \subset \Omega^2} \mathbb{E}\left[x^{(t)}_{\textsc{Br}^{(t)}(E)} \right]. \tag{1}$$

Thus, Stackelberg is defined as follows.

Definition 2.4 (Stackelberg regret)**.** Given a mechanism M and agent's policy P, suppose that the agent submits solution a_t at each round t. Then, Stackelberg regret is defined by

$$\textsc{Reg}_{M,P}(T) = T \cdot \textsc{Opt} - \sum_{t=1}^{T} x^{(t)}_{a_t}.$$

Furthermore, we define a worst-case Stackelberg by maximizing over the agent's policy, $\textsc{WReg}_M(T) = \max_{P \in \mathcal{P}} \textsc{Reg}_{M,P}(T)$. Let $\mathcal{P}_\varepsilon$ be a family of policy under which the agent always plays ε-best response. Then, we define $\textsc{WReg}_M(T, \varepsilon) = \max_{P \in \mathcal{P}_\varepsilon} \textsc{Reg}_{M,P}(T)$. If the agent is myopic, we abuse $\textsc{Reg}_M(T) = \textsc{WReg}_M(T, 0)$ to denote its worst-case Stackelberg regret.

3 Deterministic Setting

We start with a simple setting in which the agent is myopic and the utility is deterministic. In this case, the principal needs to learn the optimal eligible set, without regard to the agent's strategy. In this case, for notational simplicity, we drop the superscript (t) since the utility remains the same across the rounds. Our main result with myopic agent is presented as follows.

Theorem 3.1. *There exists a mechanism with $\textsc{Reg}(T) = O(\min(K, \log\log T))$ against myopic agent.*

The proof is based on two algorithms, one of which relies on a novel connection between our problem and the repeated posted-price mechanism problem (henceforth RPPM) by (Kleinberg and Leighton 2003), and the other is a simple algorithm that iteratively finds a (slightly) better solution. In the former, we mainly construct a reduction from our problem to RPPM, and thus recovers the regret bound $\log \log T$ of (Kleinberg and Leighton 2003).[6] In the latter algorithm with regret bound $O(K)$, the algorithm iteratively updates the eligible set so that it *excludes* at least one suboptimal solution at each round, until there's no eligible solution. Formal definition of RPPM, pseudocode of the algorithms, and the proof can be found in the appendix.

This result implies an intimate connection between our problem and RPPM, however, we observe that this does not hold beyond this simplistic setting. In fact, the utilities are always linearly negatively correlated in RPPM, on the other hand, in our setting they can be arbitrarily correlated. Moreover, the agent has multiple solutions to choose compared to only two actions of accept or reject in RPPM.

Strategic Agent

Next, we consider a more challenging scenario in which the agent tries to strategize over the rounds. Since we cannot assume that the agent will truthfully best respond to the mechanism at each iteration, instead, he possibly tries to deceive the mechanism by untruthfully submitting a solution. Thus, we need to design a mechanism that is robust to the strategic behavior. This may indeed be plausible in practice, for instance in our online labor market example, the platform may try to deceive the task requester to not strain highly qualified workers. Intuitively, this will be especially true when the platform does not have a large number of workers.

Mainly, we characterize the regret upper bound with respect to two types of assumptions. The first version of the results relies on a relatively simple assumption such that the agent's utility is uniformly bounded below by some constant. The latter depends on the Lipschitz continuity of the utility across the solutions and the density of solutions in the utility space. For each setting, we provide regret upper bounds and matching lower bounds. This justifies the necessity of the assumptions imposed, thereby characterizing the regimes in which the principal can attain large utility.

Before presenting the results, we introduce a notion of delayed mechanism, which will be useful in dealing with strategic agent. Formally, we say that mechanism M is D-delayed, if at each round t, it uses $H_{\max(1,t-D)}$ to decide its eligible set E_t. Delayed mechanism effectively restricts the strategic agent's behavior, as follows.

Lemma 3.2 ((Haghtalab et al. 2022)). *Given $\gamma \in (0,1)$, if we set $D = \lceil T_\gamma \log(T_\gamma/\varepsilon) \rceil$, then D-delayed mechanism M satisfies $\mathrm{WREG}_M(T) \leq \mathrm{WREG}_M(T, \varepsilon)$.*

Intuitively, if D gets larger enough, the agent with discounted utility is less incentivized to deviate from the best response at each round since the discounted utility after D rounds may not be enough to make up for the loss of ε utility in the current turn.

Uniformly bounded agent utility. Formally, we first assume that $Y_a > y_{\min}$ for any $a \in A$ and the principal is also aware of this lower bound. Our regret bound will accordingly be parameterized with respect to $y_{\min}$. This assumption is plausible since in our online labor market example, the task requester and the worker are typically contracted to pay an intermediary fee to the platform and thus constitute a reasonable amount of minimum payoff to the agent. The existence of such a minimum utility effectively allows us to compute the necessary delay to make the agent approximately myopic, thanks to Theorem 3.2.

Leveraging the minimum utility of the agent, we consider a variant of the algorithm used in the myopic deterministic setting, by introducing a delay in reacting to the agent's feedback. Then, we can obtain the following regret bound.

Theorem 3.3. *There exists an algorithm with $\mathrm{WREG}(T) = O(KT_\gamma \log \frac{T_\gamma}{y_{\min}})$ against γ-discounting strategic agent.*

Essentially, the delay introduced in the algorithm induces the agent to behave *restrictively strategic*, and we can effectively bound the regret to be constant, assuming the other parameters are constants. Note, however, that the regret bound linearly depends on the number of agent's solutions K. Obviously, if K tends to be large in some cases, our regret guarantee here is doomed to be pessimistic. This is indeed plausible in practice, since the agent may have growing number of solutions with respect to T, especially for online platforms.

This limitation can be handled by shrinking the eligible sets more in an aggressive manner, instead of sequentially seeking the next-best solutions. Then, the linear dependency on K can further be wiped out as follows.

Theorem 3.4. *There exists an algorithm with $\mathrm{WREG}(T) = O(T_\gamma \log \frac{T_\gamma}{y_{\min}} + \log T)$ against γ-discounting strategic agent.*

Note that the regret no longer depends on the number of solutions K, but instead on $\log T$. Our algorithm keeps shrinking the eligible set until it concludes that the truly optimal solution lies within at most $1/T$ to the currently best solution. Afterward, the regret is at most $1/T \cdot T$, thus does not affect the overall regret upper bound. Intuitively, to remove the dependence on the number of solutions, such a logarithmic burden on T is essential to guarantee that our eventual solution is correct up to $O(1/T)$ distance.

We further note that both the regret bounds of Theorem 3.3 and 3.4 have a logarithmic dependency on $1/y_{\min}$. Assuming that $T_\gamma = 1/(1-\gamma) = O(1)$, this regret bound yields a sublinear regret upper bound if $y_{\min} = e^{-o(T)}$, but becomes detrimental the other way around. Interestingly, however, we show that this dependency is necessary to obtain sublinear regret for any algorithm, by formally proving that no algorithm can achieve sublinear regret if $y_{\min} \leq e^{-T}$ against the strategic agent.

Theorem 3.5. *If $y_{\min} \leq e^{-T}$, then any algorithm has $\mathrm{WREG}(T) = \Omega(T)$ against γ-discounting strategic agent.*

Thus, the principal suffers a large amount of regret by delegating to the agent whose utility tends to be very small.

[6] Note that any state-of-the-art result can be carried over to our problem's regret bound, due to our reduction.

Lipschitz utility with dense solutions. Next, we consider the case where the principal is not aware of any lower bound on $y_{\min}$, or such a lower bound does not exist. To cope with this lack of information on $y_{\min}$, we assume that there is no significant disparity in the utility between two close solutions for both the principal and the agent, and the solutions are densely spread in the utility space. Under these assumptions, we provide an algorithm to find a near optimal solution.

These assumptions are formally presented as follows.

Assumption 3.6 (d-dense)**.** *Let $d_X(a,b) = |X_a - X_b|$. A problem instance is d-dense for some $d > 0$ if for any two solutions $a, b \in A$, either of the following is satisfied: (i) $d_X(a,b) \leq d$ or (ii) $d_X(a,b) > d$ and there exists another solution c such that $d_X(a,c) \leq d$ and $d_X(b,c) \leq d_X(a,b)$.*

Assumption 3.7 (L_1, L_2-Lipschitz continuity)**.** *There exists absolute constants $L_1, L_2 > 0$ such that for any $a, b \in A$, we have $L_1 \cdot d_X(a,b) \leq d_Y(a,b) \leq L_2 \cdot d_X(a,b)$, where $d_X(a,b) = |X_a - X_b|$ and $d_Y(a,b) = |Y_a - Y_b|$.*

Our assumption of densely spread solutions is innocuous since the solutions will be packed more in a compact manner as the number of solutions grow. Otherwise, if the number of solutions is relatively small, then our results on the bounded agent's utility would kick-in, and thus one may recover the sublinear regret. The Lipschitz continuity assumption is often valid, as it is observed that when the agent's utility for two solutions is similar, the principal's utility follows suit, and vice versa. For instance, if all the solutions lie in $y = 1 - x$, then the Lipschitz condition holds with (L_1, L_2) being $(1, 1+\varepsilon)$ or $(1-\varepsilon, 1)$ for any choice of $\varepsilon \geq 0$.

Further, we assume that Lipschitz parameters L_1 and L_2 tend to be close to each other, precisely, $L_2 \geq L_1 > \frac{3}{4}L_2$. Indeed, if there exists a significant difference between L_1 and L_2, the Lipschitz assumption fails to effectively impose any restrictions.

Leveraging these assumptions, we propose a new algorithm. Since we lack precise information on the required delay to ensure the submission of a solution, we cannot compel the agent to be approximately myopic. We propose a modified version of the algorithm used above which effectively leverages the assumptions above to explore superior solutions.

Theorem 3.8. *If $\alpha := L_1 - \frac{3}{4}L_2 > 0$, then Algorithm 1 has $\mathrm{WReg}(T) = O(T_\gamma \log \frac{T_\gamma}{\alpha} + \log \frac{1}{d} + dT)$ against γ-discounting strategic agent.*

In Algorithm 1, we maintain an interval, denoted as $[l, r]$, which encompasses the optimal solution. It is guaranteed that at every round, a solution a exists such that $X_a = l$. According to the Lipschitz continuity assumption, we can place an upper bound on r, signifying that the optimal solution a^* cannot be significantly distant from a. This is because when the difference between X_{a^*} and X_a becomes large, it is expected that $Y_{a^*} - Y_a$ will also be substantial. This is not possible since Y_{a^*} is non-negative, and cannot be significantly greater than Y_a, otherwise, it would have been proposed by the agent in earlier rounds.

With the bounded value of r, our objective is to determine if there exists a solution within the right half of the interval.

Algorithm 1: DelayedProgeressiveSearch

```
while any solution has not been received do
    Announce $E_0$.
end
Let $a_0$ be the proposed solution.
$\alpha \leftarrow L_1 - \frac{3}{4}L_2, l \leftarrow X_{a_0}, y \leftarrow Y_{a_0}$,
    $r \leftarrow \min\{1, l + \frac{y}{L_1}\}, \varepsilon \leftarrow 4\alpha d, D \leftarrow T_\gamma \log \frac{T_\gamma}{\varepsilon}$;
Announce $E_0$ for $D$ rounds.
while $r - l > 4d$ do
    $\tau \leftarrow \frac{l+r}{2}$;
    Announce $E^{>}_{\tau}$.
    if solution $a$ is proposed by the agent then
        $l \leftarrow X_a, y \leftarrow Y_a$ else $r \leftarrow \tau$ ;
    Announce $E_l$ for $D$ rounds.
    $r \leftarrow \min\{r, l + \frac{y}{L_1}\}$;
end
Announce $E_l$ for remaining rounds.
```

By considering the line $x = \frac{l+r}{2}$, the d-dense assumption implies that if a solution exists in the right half, there must be a solution with the principal's utility ranging from $\frac{l+r}{2}$ to $d + \frac{l+r}{2}$. Utilizing the Lipschitz continuity along with the condition on its parameters, we can find a lower bound on the agent's utility within that interval. Consequently, we can introduce an appropriate delay to compel the agent to propose a solution from the right half if it exists. As a result, the algorithm can determine the presence or absence of a solution in the right half and subsequently shrink the interval accordingly. By continuing this procedure, the interval gradually converges toward the optimal solution.

Furthermore, our regret upper bound essentially decomposes T_γ from T, and thus the effect of discount factor is decoupled from the linear dependency of dT. The linear dependency on dT may look pessimistic at first glance, but we reveal that this dependency is indeed optimal, as formally presented as follows.

Theorem 3.9. *There exists a d-dense problem instance such that any algorithm suffers $\mathrm{WReg}(T) = \Omega(dT)$ against γ-discounting strategic agent.*

Its proof easily follows from the proof of Theorem 3.5. Thus, this demonstrates the fundamental inevitability of the term dT. It's worth noting that whenever d is subconstant, *e.g.,* $d = T^{-c}$ for $c > 0$, then our regret upper bound in Theorem 3.8 implies a sublinear regret.

In our online labor market example, since the number of workers in a platform usually grows with respect to the time horizon, their intrinsic qualities might lie more compactly in the utility space as time flows. For instance, if there are T^ε workers having uniformly distributed utility in a compact utility space for some $\varepsilon \in (0,1)$, their utility will be $O(T^{-\varepsilon})$-densely spread, which would yield $dT = T^{1-\varepsilon} = o(T)$ regret bound. In words, the requester needs to delegate to a platform with a large number of solutions, *i.e.,* delegating to *big business matter*. Conversely, the platform should maintain more workers to attract requesters, *i.e., economy of scale* works, however, a trade-off arises since the platform's gains from strategizing would be limited then.

4 Stochastic Setting

For the stochastic setting, previous algorithms no longer work as the ex-post optimal solution varies across the rounds. Thus, the objective of the principal here will not be to find the largest threshold τ to exclude any ex-post suboptimal solutions, but to balance the probability that the agent possesses a solution that belongs to the eligible set and the corresponding utility of the agent's best response therein. Recall that this phenomenon is already well-captured in our benchmark (1) and the corresponding notion of Stackelberg regret. Given the differences, the principal faces an additional challenge of handling the random noise in the reward, and needs to find the best threshold to balance the trade-off presented above.

To cope with it, we reduce our problem to a stochastic bandit problem which is standard in the literature (Kleinberg and Leighton 2003; Amin, Rostamizadeh, and Syed 2013; Haghtalab et al. 2022). In the stochastic multi-armed bandit problem, a principal has a set of Q arms, indexed by $i \in [Q]$, and needs to decide which arm to pull at each round given the time horizon T. Each arm i is equipped with a reward distribution D_i with support $[0, 1]$. Given $\mu_i = \mathbb{E}_{r\sim D_i}[r]$, the principal's objective is to minimize expected regret defined by $\textsc{Reg}(T) = T \cdot \max_{i\in[Q]} \mathbb{E}_{r\sim D_i}[r] - \sum_{t=1}^{T} \mathbb{E}[r_t]$, where r_t denotes the random reward of the arm selected at round t by the principal.

We first discretize the space of principal's utility into the set of i/Q for $i \in [Q]$ for some carefully chosen parameter Q. Each element i/Q corresponds to a single arm, which represents the threshold τ that the principal can commit to at each round. By pulling the arm i, the principal is essentially announcing an eligible set of $E_{\tau_i} = \mathbb{1}\{a \mid X_a \geq i/Q\}$. Namely, the principal aims to find the best eligible set among the set of E_{τ_i} for $i \in [Q]$. If the discretization is dense enough with respect to the problem parameters, the regret bound here would imply a reasonable regret bound for our original problem. We define $f(\tau) = \mathbb{E}\left[x^{(t)}_{\textsc{Br}^{(t)}(E_\tau)}\right]$ as the expected principal utility for using threshold τ. Note that if the agent best responds, the expected utility from pulling arm i becomes $f(i/Q)$.

We assume that $f(\tau)$ achieves its maximum for a unique $\tau^* \in (0, 1)$ with $f''(\tau^*) < 0$, which is common in the literature (Kleinberg and Leighton 2003; Amin, Rostamizadeh, and Syed 2013; Haghtalab et al. 2022). Now, we can simply use the well-known UCB upon the discretization, and obtain the following results against the myopic agent.

Theorem 4.1. *If the agent is myopic, running UCB1 with discretization by $Q = (\frac{T}{\log T})^{1/4}$ has $\textsc{Reg}(T) = O(\sqrt{T \log T})$.*

Its analysis is a simple variant of (Kleinberg and Leighton 2003), but we provide the entire proof to make paper self-contained.

Next, to deal with the strategic behavior of the agent, we again exploit the concept of delay to restrict the agent to be approximately best responding with a suitable choice of parameters. In addition, however, we cannot simply expect that the outcome of pulling a single arm, *i.e.,* a specific eligible set, follows some stochastic distributions since the agent may strategically deviate from the best response at hand. To cope with this additional challenge, we use the foundation of perturbed bandit instance by (Haghtalab et al. 2022).

Since the stochastic setting is a generalization of the deterministic setting, it is obvious that we need a reasonable set of assumptions to obtain positive results. Similar to the deterministic setting, we first assume that there exists a value $y_{min} > 0$ such that for any realization of the agent's solutions, his utility for each solution is strictly greater than y_{min}. Secondly, we assume that the problem instance satisfies the following assumption, which is a stochastic version of Lipschitz continuity in the deterministic setting.

Definition 4.2 (Stochastic Lipschitz continuity)**.** Under the stochastic setting, we say that the problem instance is stochastically Lipschitz-continuous with parameter $L_1 > 0$ if the ex-post utilities of the solutions are correlated in a sense that for any $a, b \in A - \{a_0\}$, we have $L_1 \cdot d_X(a, b) \leq d_Y(a, b)$.

Finally, our main result can be presented as follows.

Theorem 4.3. *Under the two assumptions presented above, there exists an algorithm that has $\text{W}\textsc{Reg}(T)$ of*

$$O\left(\sqrt{T\log T} + T_\gamma \log\left(T_\gamma \max\left(\frac{T}{L_1}, \frac{1}{y_{\min}}\right)\right)\log T\right),$$

with γ-discouting strategic agent.

Our proof essentially relies on a construction of proper random perturbation interval, followed by the regret analysis of delayed version of successive elimination algorithm by (Haghtalab et al. 2022) and (Lancewicki et al. 2021). The technical subtlety lies on introducing a proper random perturbation interval to convert it to the perturbed bandit instance. The latter term including T_γ, L_1 and $y_{\min}$ incurs due to the strategic behavior of the agent. Still, this only contributes $\log T$ amount of regret once all these parameters are constants, which is dominated by the former term of $\sqrt{T \log T}$.

5 Conclusion

We study a novel *repeated delegated choice* problem. This is the first to study the online learning variant of delegated choice problem by (Armstrong and Vickers 2010; Kleinberg and Kleinberg 2018; Hajiaghayi, Rezaei, and Shin 2023). We thoroughly investigate two problem dimensions regarding whether the agent strategizes over the rounds or not, and whether the utility is stochastic or deterministic. We obtain several regret upper bounds for each problem setting, along with corresponding lower bounds that complement the hardness of the problems and some assumptions therein. Our analysis mainly characterizes the conditions of problem instances on which the principal can efficiently learn to delegate compared to the case when she knows the optimal delegation mechanism in hindsight, thereby providing fruitful insights in the principal's decision-making in delegation process.

Acknowledgements

This work is partially supported by DARPA QuICC, NSF AF:Small #2218678, NSF AF:Small #2114269 and Army-Research Laboratory (ARL) #W911NF2410052.

References

Alonso, R.; and Matouschek, N. 2008. Optimal delegation. *The Review of Economic Studies*, 75(1): 259–293.

Amador, M.; and Bagwell, K. 2013. The theory of optimal delegation with an application to tariff caps. *Econometrica*, 81(4): 1541–1599.

Amin, K.; Rostamizadeh, A.; and Syed, U. 2013. Learning prices for repeated auctions with strategic buyers. *Advances in Neural Information Processing Systems*, 26.

Armstrong, M.; and Vickers, J. 2010. A model of delegated project choice. *Econometrica*, 78(1): 213–244.

Babaioff, M.; Blumrosen, L.; Dughmi, S.; and Singer, Y. 2017. Posting prices with unknown distributions. *ACM Transactions on Economics and Computation (TEAC)*, 5(2): 1–20.

Bai, Y.; Jin, C.; Wang, H.; and Xiong, C. 2021. Sample-efficient learning of stackelberg equilibria in general-sum games. *Advances in Neural Information Processing Systems*, 34: 25799–25811.

Bendor, J.; Glazer, A.; and Hammond, T. 2001. Theories of delegation. *Annual review of political science*, 4(1): 235–269.

Birmpas, G.; Gan, J.; Hollender, A.; Marmolejo, F.; Rajgopal, N.; and Voudouris, A. 2020. Optimally deceiving a learning leader in stackelberg games. *Advances in Neural Information Processing Systems*, 33: 20624–20635.

Guo, Y.; and Hörner, J. 2021. Dynamic allocation without money.

Haghtalab, N.; Lykouris, T.; Nietert, S.; and Wei, A. 2022. Learning in Stackelberg Games with Non-myopic Agents. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, 917–918.

Hajiaghayi, M.; Rezaei, K.; and Shin, S. 2023. Multi-agent Delegated Search. *arXiv preprint arXiv:2305.03203*.

Holmstrom, B. 1980. On the theory of delegation. Technical report, Discussion Paper.

Kleinberg, J.; and Kleinberg, R. 2018. Delegated search approximates efficient search. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, 287–302.

Kleinberg, R.; and Leighton, T. 2003. The value of knowing a demand curve: bounds on regret for online posted-price auctions. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, 594–605.

Kleiner, A. 2022. Optimal Delegation in a Multidimensional World. *arXiv preprint arXiv:2208.11835*.

Lancewicki, T.; Segal, S.; Koren, T.; and Mansour, Y. 2021. Stochastic multi-armed bandits with unrestricted delay distributions. In *International Conference on Machine Learning*, 5969–5978. PMLR.

Lauffer, N.; Ghasemi, M.; Hashemi, A.; Savas, Y.; and Topcu, U. 2022. No-Regret Learning in Dynamic Stackelberg Games. *arXiv preprint arXiv:2202.04786*.

Lewis, T. R. 2012. A theory of delegated search for the best alternative. *The RAND Journal of Economics*, 43(3): 391–416.

Li, J.; Matouschek, N.; and Powell, M. 2017. Power dynamics in organizations. *American Economic Journal: Microeconomics*, 9(1): 217–241.

Lipnowski, E.; and Ramos, J. 2020. Repeated delegation. *Journal of Economic Theory*, 188: 105040.

Marecki, J.; Tesauro, G.; and Segal, R. 2012. Playing repeated stackelberg games with unknown opponents. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, 821–828.

Nisan, N.; Roughgarden, T.; Tardos, E.; and Vazirani, V. V. 2007. *Algorithmic game theory*. Cambridge university press.

Rahmani, M.; and Ramachandran, K. 2016. Dynamics of delegated search. Technical report, Working paper, Scheller College of Business, Georgia Institute of Technology.

Samuel-Cahn, E. 1984. Comparison of threshold stop rules and maximum for independent nonnegative random variables. *the Annals of Probability*, 1213–1216.

Von Stackelberg, H. 2010. *Market structure and equilibrium*. Springer Science & Business Media.

Xiao, Y.; Hu, Z.; and Wang, S. 2022. Information Design of a Delegated Search. *Available at SSRN 4249165*.

Zhao, G.; Zhu, B.; Jiao, J.; and Jordan, M. I. 2023. Online Learning in Stackelberg Games with an Omniscient Follower. *arXiv preprint arXiv:2301.11518*.