Preference-Based Distributed Welfare Maximization: A Game-Theoretic Approach

Antoine Bergerault[†] ANTOINE.BERGERAULT@UZH.CH
Anna Maddux[*] ANNA.M.MADDUX@GMAIL.COM
Andreas Schlaginhaufen[*] ANDREAS.SCHLAGINHAUFEN@EPFL.CH
Maryam Kamgarpour MARYAM.KAMGARPOUR@EPFL.CH
Sycamore, EPFL

Abstract

We consider the problem of learning to optimize a welfare objective in a multi-agent system coordinated by a central authority. This setting presents two main challenges: (i) the welfare function is unknown or difficult to specify explicitly, and (ii) centralized optimization is intractable due to the exponential dependence on the number of agents. We address these challenges by combining preference-based learning with a game-theoretic reformulation of the central optimization problem. By designing agents' utilities aligned with the social welfare, this formulation enables independent learning to maximize the welfare value. Specifically, we propose a novel algorithm that iteratively combines dueling bandit-style preference learning with game-theoretic no-regret learning to guide agents' actions. Under a submodularity assumption on the welfare function, we prove that our proposed algorithm has sublinear regret. Our regret guarantee furthermore implies that, with high probability, the average welfare over T rounds is near-optimal up to a constant depending on the curvature of the welfare function. Finally, we validate our approach in a case study on rebalancing a shared mobility system, where vehicles are placed strategically across different areas.

Keywords: Online learning, preference-based optimization, game theory.

1. Introduction

Resource allocation is a fundamental challenge in many real-world systems, from electricity markets [10] and sensor placement [19] to shared mobility systems [48]. In these settings, a central authority distributes resources across various locations subject to capacity constraints, aiming to maximize a *welfare function* that quantifies the quality of the allocation. However, in many applications, the welfare function is not a single easy-to-interpret metric, but a complex trade-off between competing goals such as user satisfaction, long-term efficiency, fairness, and operational costs [14, 20].

Consequently, formalizing an explicit welfare function is difficult, and it is unrealistic to assume the central authority can query the welfare value of an executed allocation. However, the welfare value might be available through human feedback, such as user satisfaction [16]. Thus, a more practical and weaker assumption is that the central authority can query preferences between pairs of allocations executed online based on their outcomes. This transforms the welfare maximization problem into an online dueling bandit problem [2, 46].

[†] Work done while at EPFL. Now at the University of Zurich.
[*] Equal contribution.

© A. Bergerault, A. Maddux, A. Schlaginhaufen & M. Kamgarpour.

To tackle optimization from preference feedback, past work [2, 13, 24] reduces the problem to a stochastic bandit setting [1], applying the classical upper confidence bound algorithm [4] to minimize regret. However, these methods rely on a central optimization oracle to maximize the optimistic welfare estimates over the set of all possible allocations. This cannot be efficiently applied to large-scale resource allocation problems whose set of possible allocations grows exponentially with the number of resources and locations, e.g. the numerous charging stations in station-based mobility systems or the possible locations when placing sensors across a large facility.

To address this computational bottleneck, we adopt a distributed optimization perspective. Following the game-theoretic formulation of Marden and Wierman [27], we model the optimization problem as a repeated game distributing the computational burden across multiple agents, each acting on a subset of the action space. To ensure convergence to a high welfare value, we design agents' utilities to align with the welfare. We build on previous work on distributed welfare maximization in the bandit setting [38, 43], that we extend to the case of preference feedback using recent advances in (preference-based) online learning [24, 35]. More information on related work in Appendix A.

Contributions Our contributions are threefold: First, we propose Distributed Optimization via Preference-based Learning (DO-PL), a distributed algorithm for preference-based optimization of an unknown welfare function. Second, under submodularity and linearity assumptions, we prove that DO-PL converges to a time-averaged welfare at least a $1/2$-approximation of the global optimum. Finally, we validate our approach in a numerical case study of a shared mobility system.

2. Problem Setup

We consider a repeated resource allocation problem where a central authority coordinates a system of N agents over a horizon of T rounds. The system's joint allocation space $\mathcal{A}$ is the Cartesian product of N individual subsets, i.e. $\mathcal{A} = \mathcal{A}^1 \times \cdots \times \mathcal{A}^N$, where each $\mathcal{A}^i$ represents the feasible allocations for agent i. At each round t, the central authority selects a joint allocation $a_t \in \mathcal{A}$, which is executed in the environment and induces an outcome.

The authority's goal is to maximize a global welfare function $\gamma : \mathcal{A} \to \mathbb{R}$, which typically measures a complex trade-off of metrics such as user satisfaction, revenue, and operational risks. The objective is to select a sequence of allocations $\{a_t\}_{t=1}^T$ that minimizes the average regret $\frac{1}{T}\sum_{t=1}^T (\mathrm{OPT} - \gamma(a_t))$ relative to the optimal allocation $\mathrm{OPT} = \max_{a\in\mathcal{A}} \gamma(a)$.

In many applications, this γ is difficult to formalize and only accessible via preference feedback. To optimize it, we therefore make the following two popular modeling assumptions [12, 26, 34, 41].

Assumption 1 (Bradley–Terry model) *The probability that allocation a is preferred over $\tilde{a}$, denoted as $a \succ \tilde{a}$, is given by $\mathbb{P}(a \succ \tilde{a}) = \sigma(\gamma(a) - \gamma(\tilde{a}))$, where $\sigma(x) = 1/(1+e^{-x})$. We record the feedback as a tuple (a^+, a^-), where $a^+ \succ a^-$.*

Assumption 2 (Linearity) *The welfare function satisfies $\gamma(a) = \gamma_{\theta^*}(a) := \langle \theta^*, \phi(a)\rangle$, for a feature mapping $\phi : \mathcal{A} \to \mathbb{R}^d$ and an unknown $\theta^* \in \mathbb{R}^d$, with $\|\theta^*\|_2 \le B$, $\|\phi(a)\|_2 \le L$ and $d \in \mathbb{Z}^+$.*

Under this setup, we describe our distributed algorithm in the next section.

3. A Distributed Algorithm for Preference-Based Optimization

We formulate the optimization of the unknown welfare function γ as a repeated game over T rounds. We exploit the decoupled structure of the allocation set $\mathcal{A} = \mathcal{A}^1 \times \ldots \times \mathcal{A}^N$ by distributing the computational burden across N agents. Specifically, at each round t, the central authority assigns each agent i a local utility function $U_t^i : \mathcal{A} \to \mathbb{R}$, which induces a game among the agents. Agents then independently select actions $a_t^i \in \mathcal{A}^i$ to maximize their respective utilities. Finally, the central authority expresses its preference over the current action profile a_t and a reference action profile $\tilde{a}_t$.

Distributedly optimizing an unknown welfare function presents two primary challenges: (i) the central authority must learn γ using only comparative preference feedback, and (ii) it must design agents' utility functions such that individual optimization aligns with the global maximization of γ.

3.1. Estimating the Welfare Function from Preference Feedback

To estimate the unknown welfare function γ, we combine the preference model in Assumption 1 with the linearity assumption in Assumption 2. Under these assumptions, we estimate the parameter θ^* using *maximum likelihood estimation* (MLE) based on the collected preference dataset $\mathcal{D}_t = \{(a_\tau^+, a_\tau^-)\}_{\tau=1}^{t-1}$. The constrained MLE is given by $\hat{\theta}_t = \arg\min_{\|\theta\|_2 \leq B} \mathcal{L}_{\mathcal{D}_t}(\theta)$ where $\mathcal{L}_{\mathcal{D}_t}(\theta)$ is the standard logistic loss[1] [39] $\mathcal{L}_{\mathcal{D}_t}(\theta) = -\sum_{(a^+,a^-)\in\mathcal{D}_t} \log \sigma\left(\langle \theta, \phi(a^+) - \phi(a^-)\rangle\right)$, which can be optimized efficiently via e.g. projected gradient methods [8].

At round t, we will use $\hat{\theta}_t$ as an estimator for the unknown parameter θ^*. In particular, $\hat{\theta}_t$ allows the central authority to approximate welfare differences $\gamma(a) - \gamma(\tilde{a})$ for any pair $a, \tilde{a} \in \mathcal{A}$. The reliability of this estimation is governed by the following time-uniform confidence bound:

Lemma 1 (Lemma 3.1 in Schlaginhaufen et al. [35]) *Let Assumptions 1 and 2 hold. For any $\lambda > 0$, with probability at least $1 - \delta$ for all $a, \tilde{a} \in \mathcal{A}$ and $t \geq 1$ it holds that:*

$$\left|(\gamma(a) - \gamma(\tilde{a})) - \left(\gamma_{\hat{\theta}_t}(a) - \gamma_{\hat{\theta}_t}(\tilde{a})\right)\right| \leq \beta_t(\delta)\sigma_t(a, \tilde{a}),$$

where $\beta_t(\delta) := \mathcal{O}\left(\sqrt{\kappa\left[\log\left(\frac{1}{\delta}\right) + d\log\left(\frac{t}{d}\right)\right] + \lambda}\right)$ with $\delta \in (0,1)$, $\kappa = \mathcal{O}(e^{2BL})$, $\sigma_t(a, \tilde{a}) := \|\phi(a) - \phi(\tilde{a})\|_{V_t^{-1}}$, $V_t := \lambda I + \sum_{s=1}^{t-1} x_s x_s^\top$ and $x_s = \phi(a_s) - \phi(\tilde{a}_s)$.

As a direct consequence, we can upper-bound the difference $\gamma(a) - \gamma(\tilde{a})$ with high probability by the estimate $\gamma_{\hat{\theta}_t}(a) - \gamma_{\hat{\theta}_t}(\tilde{a}) + \beta_t(\delta)\sigma_t(a, \tilde{a})$, forming the basis for our utility design strategy.

3.2. Utility Design and Distributed Learning

To achieve global coordination, we define the following optimistic utility functions $U_t^i : \mathcal{A} \to \mathbb{R}$:

$$U_t^i(a) := \gamma_{\hat{\theta}_t}(a) - \gamma_{\hat{\theta}_t}(\tilde{a}_t) + \beta_t(\delta)\sigma_t(a, \tilde{a}_t), \tag{1}$$

where $\hat{\theta}_t$ is the MLE and $\tilde{a}_t$ a fixed reference allocation. Lemma 1 ensures that $U_t^i(a) \geq \gamma(a) - \gamma(\tilde{a}_t)$ with probability at least $1 - \delta$. While following the classical optimism principle, we focus on welfare differences as γ is only identifiable up to constant shifts from preference feedback. Since

1. This corresponds to the negative log-likelihood of the Bradley-Terry model.

any maximizer of γ also maximizes $\gamma(\cdot) - \gamma(\tilde{a}_t)$, this methodology ensures that agents' individual optimization remains aligned with the global welfare γ, shrinking uncertainty as data accumulates.

The assignment of U_t^i induces a game where, at each round t, agent i independently selects an action $a_t^i \in \mathcal{A}^i$ and receives $U_t^i(a_t^i, a_t^{-i})$, where a_t^{-i} denotes the actions of all other agents. To maximize the welfare, we will distribute the optimization by making the agents independently optimize their individual learning regrets $R^i(T) = \max_{a^i \in \mathcal{A}^i} \sum_{t=1}^T \left(U_t^i(a^i, a_t^{-i}) - U_t^i(a_t^i, a_t^{-i})\right)$, where classical no-regret algorithms ensure[2] $R^i(T)/T \to 0$ with T.

Recall that the central authority's objective is to maximize the average cumulative welfare $\frac{1}{T}\sum_{t=1}^T \gamma(a_t)$. Intuitively, since agents' utility functions are designed to be aligned with the social welfare function, assuming that all agents select an action $a_t^i \in \mathcal{A}^i$ by independently applying a no-regret algorithm ensures that this objective is approximately achieved. This relationship is formalized in our welfare guarantees in Theorem 4, which depend on the time-averaged regret.

3.3. The DO-PL Algorithm

Below we present our algorithm, Distributed Optimization via Preference-based Learning (DO-PL, Algorithm 1), that proceeds in three stages at each round t. First, preference data is used to estimate θ^* using maximum likelihood estimation (Line 9). Second, the resulting estimator is used to design optimistic individual utility functions $U_t^i : \mathcal{A} \to \mathbb{R}$ defined as in Equation (1) (Line 10). Finally, to make the algorithm efficient[3], these are optimized online via N parallel no-regret algorithms, selecting an action a_t^i for each agent (Lines 6 and 11).

Algorithm 1 Distributed Optimization via Preference-based Learning (DO-PL)

1: **Input:** Number of iterations T, confidence $\delta > 0$, regularization $\lambda > 0$
2: Initialize design matrix $V_1 = \lambda I \in \mathbb{R}^{d\times d}$ and preference dataset $\mathcal{D}_1 = \emptyset$
3: Arbitrarily initialize $\hat{\theta}_1 \in \mathbb{R}^d$, $a_0 \in \mathcal{A}$, and $U_1^i(a) : \mathcal{A} \to \mathbb{R}$ for all $i \in \mathcal{N}$.
4: $\text{Algo}^i \leftarrow \text{NO-REGRET}(\mathcal{A}^i),\ i = 1, \dots, N$
5: **for** $t = 1, \dots, T$ **do**
6: $\quad a_t^i \leftarrow \text{Algo}^i().\text{play_action}(),\ i = 1, \dots, N$
7: $\quad a_t = (a_t^1, \dots, a_t^N)$ and $\tilde{a}_t = a_{t-1}$
8: $\quad \mathcal{D}_{t+1} = \mathcal{D}_t \cup \{(a^+, a^-)\}$, where

$$(a^+, a^-) = \begin{cases} (a_t, \tilde{a}_t), & \text{if } a_t \succ \tilde{a}_t, \\ (\tilde{a}_t, a_t), & \text{otherwise.} \end{cases}$$

9: $\quad \hat{\theta}_{t+1} \in \arg\min_{\|\theta\|_2 \le B} \mathcal{L}_{\mathcal{D}_{t+1}}(\theta)$
10: $\quad U_t^i(a) = \gamma_{\hat{\theta}_t}(a) - \gamma_{\hat{\theta}_t}(\tilde{a}_t) + \beta_t(\delta)\sigma_t(a, \tilde{a}_t)$
11: $\quad \text{Algo}^i.\text{update}(a_t, U_t^i(\cdot)),\ i = 1, \dots, N$
12: **end for**

2. Such an example is the *Exponential-weight algorithm for Exploration and Exploitation with Implicit eXploration* (Exp3-IX, Lattimore and Szepesvári [22]), which under proper learning rate selection guarantees that with probability at least $1 - \delta$, $R^i(T) = \mathcal{O}\left(\sqrt{TK^i \log(K^i/\delta)}\right)$ where $K^i = |\mathcal{A}^i|$ [30].

3. DO-PL has linear computation complexity in N, compared to the exponential complexity of a centralized solution.

Throughout this process, preference feedback is collected for the selected allocation $a_t = (a_t^1, \ldots, a_t^N)$ and the comparison allocation $\tilde{a}_t = a_{t-1}$. In general, $\tilde{a}_t$ could be any comparator, but the last selected allocation a_{t-1} is a practical choice when the preferences are retrieved by asking humans about outcomes from executing $a_t, \tilde{a}_t$ in the real-world.

4. Theoretical Guarantees

While Assumptions 1 and 2 ensure the learnability of γ, an additional assumption is required to bound the suboptimality of allocations learned decentrally. This is because even if γ were known and optimized in an identical interest game, independent no-regret learning only guarantees convergence to a coarse correlated equilibrium [9], which can be highly inefficient in general [6, 33].

To address this, we leverage structural properties common in resource allocation problems: monotonicity and submodularity [31, 44]. Monotonicity implies that the marginal welfare of additional resources is non-negative, while submodularity captures the property of diminishing marginal returns [29, 40]. Importantly, submodularity guarantees that any equilibrium solution achieves at least half of the optimal welfare value [43].[4] We formalize these requirements below.

Assumption 3 (Submodularity) *Given $\mathcal{A} \subseteq \mathbb{R}_+^m$ for some $m > 0$, the welfare function satisfies:*

1. ***Monotonicity:*** *For all $a, \tilde{a} \in \mathcal{A}$ with $a \leq \tilde{a}$ element-wise, $\gamma(a) \leq \gamma(\tilde{a})$.*
2. ***DR-Submodularity:*** *For all $a, \tilde{a} \in \mathcal{A}$ with $a \leq \tilde{a}$, $k \in \mathbb{R}_+$, unit vectors $e^l \in \mathbb{R}^m$, and $l \in [m]$, such that $(a + ke^l), (\tilde{a} + ke^l) \in \mathcal{A}$, we have: $\gamma(a + ke^l) - \gamma(a) \quad \geq \quad \gamma(\tilde{a} + ke^l) - \gamma(\tilde{a})$.*
3. ***Normalization:*** *$\gamma(\mathbf{0}) = 0$ for some $\mathbf{0} \in \mathcal{A}$.*

Under Assumption 3, the work of Vetta [43] and later Sessa et al. [37] provide tighter suboptimality bounds by leveraging the notion of *curvature*. The curvature quantifies the degree to which γ deviates from linearity, capturing the severity of diminishing returns across the domain $\mathcal{A}$.

Definition 2 (Curvature [37]) *Let γ satisfy Assumption 3. For any subset $\mathcal{X} \subseteq \mathcal{A}$, the curvature of γ with respect to $\mathcal{X}$ is defined as: $\alpha(\mathcal{X}) = 1 - \inf_{\substack{a \in \mathcal{X}, \\ l \in [m]: a + ke^l \in \mathcal{X}}} \lim_{k \downarrow 0} \frac{\gamma(a+ke^l) - \gamma(a)}{\gamma(ke^l) - \gamma(0)}$.*

If γ is linear on $\mathcal{A}$, then $\alpha(\mathcal{A}) = 0$. Conversely, the monotonicity of γ ensures that $\alpha(\mathcal{A}) \in [0, 1]$. We obtain the following structural property taken from Proposition 3 in [37]:

Proposition 3 ([37]) *Let γ satisfy Assumption 3 and denote by α its curvature on $\mathcal{A}$. For any $a, \tilde{a} \in \mathcal{A}$ such that $a + \tilde{a} \in \mathcal{A}$, it holds that $\gamma(a + \tilde{a}) - \gamma(a) \geq (1 - \alpha)\gamma(\tilde{a})$.*

This lower bound enables us to characterize the suboptimality of allocations learned via DO-PL. We formalize this in Theorem 4, whose proof is deferred to Appendix B.2. We additionally adapt this result to the Exp3-IX no-regret algorithm in Appendix B.3.

4. This result holds for monotone welfare functions; in the non-monotone cases, this approximation holds up to a function-dependent additive term.

Theorem 4 *Let Assumptions 1, 2, and 3 hold and let $\delta \in (0,1)$. Then, running DO-PL (Algorithm 1) ensures that with probability at least $1-\delta$ it holds that:*

$$\frac{1}{T}\sum_{t=1}^{T}\gamma(a_t) \geq \max\left\{(1-\alpha), \frac{1}{2}\right\}\mathrm{OPT} - \sum_{i=1}^{N}\frac{R^i(T)}{T} - \frac{2N\beta_T(\delta)}{T}\sum_{t=1}^{T}\sigma_t(a_t, a_{t-1}).$$

where $\alpha = \alpha(\mathcal{A}_{aug})$ is the curvature of γ with respect to $\mathcal{A}_{aug} := \{a + \tilde{a} \mid a, \tilde{a} \in \mathcal{A}\}$.

Theorem 4 implies that the time-averaged welfare is at least $1/2$-optimal, subject to two vanishing error terms:[5] $\sum_{i=1}^{N} R^i(T)/T$, from independent no-regret learning, and $\frac{2N\beta_T(\delta)}{T}\sum_{t=1}^{T}\sigma_t(a_t, a_{t-1})$, for the statistical estimation error of γ. Consequently, after $T = \mathcal{O}(1/\epsilon^2)$ rounds, the average accumulated welfare $\frac{1}{T}\sum_{t=1}^{T}\gamma(a_t)$ is at least $\frac{1}{2}\mathrm{OPT} - \epsilon$. Thus, as $T \to \infty$, DO-PL achieves a $\bar{\alpha}$-approximation of OPT with $\bar{\alpha} \in [0.5, 1]$. For comparison, this is the same rate attained by D-SubUCB [38] for resource allocation with contextual information. While the performance matches up to constant terms, D-SubUCB directly relies on bandit feedback of the form $\gamma(a_t) + \zeta_t$, where ζ_t is σ-sub-Gaussian noise, and γ is modelled via a Reproducing Kernel Hilbert Space [36].

5. Experiments

We empirically validate our framework on the problem of rebalancing a shared mobility system, in a similar experimental setup as Sessa et al. [38]. During the day, vehicles can be freely used inside the boundaries of a city, and a central authority decides to strategically replace them at night according to some allocation $a \in \mathcal{A}$ to maximize a welfare function γ. This γ is linear in an *unknown* parameter θ^* and trades-off between the following features: number of served trips, total vehicle distance, average walking distance to the bikes, and ratio of unmet demand over total demand.

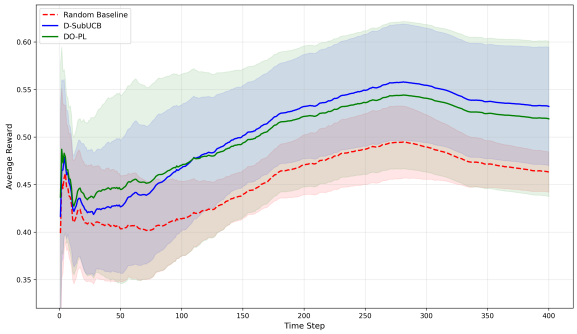

Figure 1: Average reward for the three algorithms (D-SubUCB, DO-PL, Random Baseline) over $T = 400$ steps, averaged over 100 experiments. For each experiment, we sample $\theta^* \sim \mathrm{Dirichlet}(4)$ and run the algorithms in parallel using the same source of randomness.

In Figure 1, we illustrate the performance of DO-PL in comparison to a random baseline taking actions uniformly at random, and the D-SubUCB algorithm[6] of Sessa et al. [38]. DO-PL signifi-

5. Recall that $\alpha \in [0, 1]$.
6. In comparison to our algorithm, D-SubUCB assumes that γ can be observed for the executed allocations, or, equivalently, that θ^* is *known*.

cantly outperforms the random baseline over the entire horizon. It does so, while slightly underperforming D-SubUCB that directly learns from the realized welfare values. Additional details and experiments can be found in Appendix C.

6. Conclusion

This work develops DO-PL, an algorithm for the distributed optimization of unknown objectives using pairwise preference feedback, with applications including shared-mobility systems [48], energy management [45], and sensor placement [19]. Under the assumption of a linear, monotone, and submodular welfare function, we show that asymptotically our algorithm achieves at least an $\frac{1}{2}$-approximation of OPT. We see linearity and submodularity as limiting assumptions, as they might not fully capture behaviors of complex real-world systems; extending the framework beyond those remains an open challenge. However, compared to standard distributed optimization and dueling bandits methods, our resulting algorithm is both tractable and does not require access to the underlying γ. Finally, a promising extension is to apply this approach to incentive design, where the central authority has limited flexibility in designing utilities and must instead shape them by incentivizing the rational agents via tolls or taxes.

References

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in Neural Information Processing Systems*, 24, 2011.

[2] Nir Ailon, Zohar Karnin, and Thorsten Joachims. Reducing dueling bandits to cardinal bandits. In *International Conference on Machine Learning*, pages 856–864. PMLR, 2014.

[3] Gürdal Arslan, Jason R Marden, and Jeff S Shamma. Autonomous vehicle-target assignment: A game-theoretical formulation. 2007.

[4] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

[5] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

[6] Siddharth Barman and Katrina Ligett. Finding any nontrivial coarse correlated equilibrium is hard. *ACM SIGecom Exchanges*, 2015.

[7] Dimitri Bertsekas and John Tsitsiklis. *Parallel and distributed computation: numerical methods*. Athena Scientific, 2015.

[8] Mathieu Blondel, Quentin Berthet, Marco Cuturi, Roy Frostig, Stephan Hoyer, Felipe Llinares-López, Fabian Pedregosa, and Jean-Philippe Vert. Efficient and modular implicit differentiation. *Advances in Neural Information Processing Systems*, 35:5230–5242, 2022.

[9] Avrim Blum, MohammadTaghi Hajiaghayi, Katrina Ligett, and Aaron Roth. Regret minimization and the price of total anarchy. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 373–382, 2008.

[10] Roger E Bohn, Michael C Caramanis, and Fred C Schweppe. Optimal pricing in electrical networks over space and time. *The Rand Journal of Economics*, pages 360–376, 1984.

[11] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 2011.

[12] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[13] Louis Faury, Marc Abeille, Clément Calauzènes, and Olivier Fercoq. Improved optimistic algorithms for logistic bandits. In *International Conference on Machine Learning*, pages 3052–3060. PMLR, 2020.

[14] Peter J Fleming, Robin C Purshouse, and Robert J Lygoe. Many-objective optimization: An engineering design perspective. In *International conference on evolutionary multi-criterion optimization*, pages 14–32. Springer, 2005.

[15] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[16] Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I*, pages 99–127. World Scientific, 2013.

[17] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

[18] Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability*, 2014.

[19] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(2), 2008.

[20] Karl-Heinz Küfer, Alexander Scherrer, Michael Monz, Fernando Alonso, Hans Trinkaus, Thomas Bortfeld, and Christian Thieke. Intensity-modulated radiotherapy–a large scale multi-criteria programming problem. *OR spectrum*, 25(2):223–249, 2003.

[21] Leo Landolt, Anna Maddux, Andreas Schlaginhaufen, Saurabh Vaishampayan, and Maryam Kamgarpour. Eliciting truthful feedback for preference-based learning via the vcg mechanism. *arXiv preprint arXiv:2510.17285*, 2025.

[22] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[23] Junghyun Lee, Se-Young Yun, and Kwang-Sung Jun. Improved regret bounds of (multinomial) logistic bandits via regret-to-confidence-set conversion. In *International Conference on Artificial Intelligence and Statistics*, pages 4474–4482. PMLR, 2024.

[24] Junghyun Lee, Se-Young Yun, and Kwang-Sung Jun. A unified confidence sequence for generalized linear models, with applications to bandits. *Advances in Neural Information Processing Systems*, 37:124640–124685, 2024.

[25] Ruicheng Liu, Jianyu Xu, Çağatay Iris, and Jianghang Chen. Dynamic rebalancing strategies for dockless bike-sharing systems. *International Journal of Production Economics*, page 109634, 2025.

[26] R Duncan Luce et al. *Individual choice behavior*, volume 4. Wiley New York, 1959.

[27] Jason R Marden and Adam Wierman. Distributed welfare games. *Operations Research*, 2013.

[28] Alberto Maria Metelli, Simone Drago, and Marco Mussi. A novel self-normalized bernstein-like dimension-free inequality and regret bounds for generalized kernelized bandits. In *Eighteenth European Workshop on Reinforcement Learning*, 2025.

[29] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14:265–294, 1978.

[30] Gergely Neu. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. *Advances in Neural Information Processing Systems*, 28, 2015.

[31] Dario Paccagnan and Jason R Marden. Utility design for distributed resource allocation—part ii: Applications to submodular, covering, and supermodular problems. *IEEE Transactions on Automatic Control*, 67(2):618–632, 2021.

[32] Barna Pásztor, Parnian Kassraie, and Andreas Krause. Bandits with preference feedback: A stackelberg game perspective. *Advances in Neural Information Processing Systems*, 37: 11997–12034, 2024.

[33] Tim Roughgarden. Intrinsic robustness of the price of anarchy. *Journal of the ACM (JACM)*, 62(5):1–42, 2015.

[34] Aadirupa Saha, Tomer Koren, and Yishay Mansour. Dueling convex optimization. In *International Conference on Machine Learning*, pages 9245–9254. PMLR, 2021.

[35] Andreas Schlaginhaufen, Reda Ouhamma, and Maryam Kamgarpour. Efficient preference-based reinforcement learning: Randomized exploration meets experimental design. *Advances in Neural Information Processing Systems*, 2025.

[36] Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In *International conference on computational learning theory*, pages 416–426. Springer, 2001.

[37] Pier Giuseppe Sessa, Maryam Kamgarpour, and Andreas Krause. Bounding inefficiency of equilibria in continuous actions games using submodularity and curvature. In *International Conference on Artificial Intelligence and Statistics*, pages 2017–2027. PMLR, 2019.

[38] Pier Giuseppe Sessa, Ilija Bogunovic, Andreas Krause, and Maryam Kamgarpour. Online submodular resource allocation with applications to rebalancing shared mobility systems. In *International Conference on Machine Learning*, pages 9455–9464. PMLR, 2021.

[39] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[40] Matthew Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. *Advances in Neural Information Processing Systems*, 21, 2008.

[41] Tomasz Strzalecki. Stochastic choice theory. *Cambridge Books*, 2025.

[42] John N Tsitsiklis. Problems in decentralized decision making and computation. Technical report, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1984.

[43] Adrian Vetta. Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.* IEEE, 2002.

[44] Jan Vondrák. *Submodularity in combinatorial optimization*. PhD thesis, Univerzita Karlova, Matematicko-fyzikální fakulta, 2007.

[45] Wenjie Xu, Wenbin Wang, Yuning Jiang, Bratislav Svetozarevic, and Colin N. Jones. Principled preferential bayesian optimization. In *International Conference on Machine Learning*. JMLR.org, 2024.

[46] Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *International Conference on Machine Learning*, pages 1201–1208, 2009.

[47] Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

[48] Rick Zhang and Marco Pavone. Control of robotic mobility-on-demand systems: a queueing-theoretical perspective. *The International Journal of Robotics Research*, 35(1-3):186–203, 2016.

Camera-Ready Edits Summary

Beyond using the camera-ready template, we only performed small modifications compared to the initially submitted paper.

Specifically, we did the following.

- Added topic keywords after the abstract.
- Fixed small typos throughout the main text.
- Fixed small typos in the proof of Theorem 4 (Appendix B.2), with no impact on the final results.

None of the conclusions or contributions of the work were impacted.

Appendix A. Related work

In the following, we review the distributed optimization and preference-based learning literature, as these areas constitute the primary building blocks for the development of our framework.

Distributed Optimization Distributed optimization decomposes a global objective into sub-problems solved via the localized actions of independent agents. These approaches are motivated by the need to preserve data privacy, satisfy communication constraints [17], and alleviate the computational bottlenecks inherent in centralized systems [42]. Conventional methods, such as the Alternating Direction Method of Multipliers (ADMM, see e.g. Boyd et al. [11]) and consensus-based optimization [7], rely on algorithmic protocols, where agents follow iterative update rules, such as gradient steps or neighbor-averaging, to collectively converge to a global solution.

In contrast, the game-theoretic approach to distributed optimization [3, 27] shifts the focus from algorithmic coordination to *utility design*. Rather than prescribing specific computational steps, this framework involves designing local utility functions that ensure individual agent incentives are aligned with the global welfare objective.

Although Sessa et al. [38] extended this framework to unknown welfare functions using a learning-based approach, their methodology relies on *bandit feedback*. This requires observing welfare values to provide convergence guarantees [18]. Our work departs from this by addressing settings where only pairwise preference feedback is available, representing a weaker and more practical feedback model.

Preference-Based Learning Preference-based learning was first studied under the name of dueling bandits, where numerical feedback is replaced with pairwise comparisons [2, 46, 47]. A standard modeling assumption for the preferences is the Bradley-Terry model [12, 26], which assumes an underlying objective function and maps objective value differences to stochastic preferences via a logistic link. Assuming a linear objective function, this problem reduces to online logistic regression on feature differences – a specific instance of a generalized linear bandit problem. For this setting, Faury et al. [13] and Lee et al. [23] establish time-uniform confidence sequences and prove regret bounds for optimistic algorithms. Recently, similar regret guarantees have been established for non-parametric function classes [28, 32, 45].

Learning optimal allocations from preference-based feedback has been previously explored by Landolt et al. [21]. In their work, the authors consider a central authority computing allocations to minimize agents' aggregated costs, given access to individual reported preferences. Their approach focuses on the mechanism design challenge of eliciting truthful preference feedback from self-interested agents. In contrast, we assume the planner has access to truthful preference feedback but consider more general welfare functions, and we focus on efficiency by distributing the optimization across agents' individual action spaces.

Appendix B. Analysis of submodular welfare functions

In this section, we provide formal proofs for our main welfare guarantees. Before establishing these results, we present a technical lemma that we will leverage in the subsequent analysis.

B.1. Technical background

The following lemma is a variation of the classic elliptical potential lemma [22, Lemma 19.4] using an observation of [22, Exercise 19.3] to avoid naive truncation. It provides an upper bound on the

sum of norms of sequentially observed vectors in the norm induced by their design matrix. These norms of vectors are commonly called elliptical potentials.

Lemma 5 (Elliptical lemma [35]) *Let $\{x_t\}_{t\geq 0} \in \mathbb{R}^d$ and for all $t \geq 0$, $\|x_t\| \leq L$, let $V_t = \sum_{s=1}^{t-1} x_s x_s^\top + \lambda I$ for some $\lambda > 0$. Then, it holds that:*

$$\sum_{t=1}^{T} \|x_t\|^2_{V_t^{-1}} \leq 2d \log\left(1 + \frac{TL^2}{d\lambda}\right) + \frac{3dL^2}{\log(2)\lambda} \log\left(1 + \frac{L^2}{\log(2)\lambda}\right).$$

B.2. Proof of Theorem 4

Proof The welfare γ accumulated across T rounds can be written as:

$$\begin{aligned}
\sum_{t=1}^{T} \gamma(a_t) &\overset{(i)}{=} \sum_{t=1}^{T} \left(\sum_{i=1}^{N} \gamma(a_t^{1:i}, 0) - \gamma(a_t^{1:i-1}, 0) \right) \\
&\overset{(ii)}{\geq} \sum_{t=1}^{T} \sum_{i=1}^{N} (\gamma(a_t) - \gamma(0, a_t^{-i})) \\
&= \sum_{t=1}^{T} \sum_{i=1}^{N} (\gamma(a_t) - \gamma(a_{t-1}) + \gamma(a_{t-1}) - \gamma(0, a_t^{-i})) \qquad (2)
\end{aligned}$$

where (i) uses $\gamma(0) = 0$ and (ii) is from the sub-modularity of γ.

By Lemma 1, with probability at least $1 - \delta$ for all $a, \tilde{a} \in \mathcal{A}$ it holds that:

$$\begin{aligned}
|(\gamma(a) - \gamma(\tilde{a})) - (\gamma_{\hat{\theta}_t}(a) - \gamma_{\hat{\theta}_t}(\tilde{a}))| &= |\langle \theta^* - \hat{\theta}_t, \phi(a) - \phi(\tilde{a}) \rangle| \\
&\leq \|\theta^* - \hat{\theta}_t\|_{V_t} \|\phi(a) - \phi(\tilde{a})\|_{V_t^{-1}} \\
&\leq \beta_t(\delta) \sigma_t(a, \tilde{a}).
\end{aligned}$$

This implies that for all $a \in \mathcal{A}$ the following inequalities hold:

$$\begin{aligned}
&\Leftrightarrow \gamma(a) - \gamma(a_{t-1}) \geq \gamma_{\hat{\theta}_t}(a) - \gamma_{\hat{\theta}_t}(a_{t-1}) - \beta_t(\delta)\sigma_t(a, a_{t-1}) \\
&\Leftrightarrow (\gamma(a) - \gamma(a_{t-1})) \geq U_t^i(a) - 2\beta_t(\delta)\sigma_t(a, a_{t-1}), \qquad (3)
\end{aligned}$$

where $U_t^i(a)$ is defined as in Equation (1). Plugging the above inequality into Equation (2), the following holds with probability at least $1 - \delta$:

$$\begin{aligned}
\sum_{t=1}^{T} \gamma(a_t) &\geq \sum_{t=1}^{T} \sum_{i=1}^{N} (\gamma(a_t) - \gamma(a_{t-1}) + \gamma(a_{t-1}) - \gamma(0, a_t^{-i})) \\
&\geq \sum_{t=1}^{T} \sum_{i=1}^{N} U_t^i(a_t) + \sum_{t=1}^{T} \sum_{i=1}^{N} \gamma(a_{t-1}) - \sum_{t=1}^{T} \sum_{i=1}^{N} \gamma(0, a_t^{-i}) - 2 \sum_{t=1}^{T} \sum_{i=1}^{N} \beta_t(\delta)\sigma_t(a_t, a_{t-1}).
\end{aligned}$$

Let $a^* \in \arg\max_{a \in \mathcal{A}} \gamma(a)$ and $\overline{a}^i \in \arg\max_{a^i \in \mathcal{A}^i} \sum_{t=1}^{T} U_t^i(a^i, a_t^{-i})$ for all $i \in [N]$. Then, by definition of agent i's regret, it holds that

$$R^i(T) = \sum_{t=1}^{T} \left(U_t^i(\overline{a}^i, a_t^{-i}) - U_t^i(a_t^i, a_t^{-i}) \right) \geq \sum_{t=1}^{T} \left(U_t^i(a^{*,i}, a_t^{-i}) - U_t^i(a_t^i, a_t^{-i}) \right).$$

Thus, the following holds with probability at least $1-\delta$:

$$\begin{aligned}\sum_{t=1}^{T}\gamma(a_t) &\geq \sum_{t=1}^{T}\sum_{i=1}^{N}U_t^i(a_t)+\sum_{t=1}^{T}\sum_{i=1}^{N}\gamma(a_{t-1})-\sum_{t=1}^{T}\sum_{i=1}^{N}\gamma(0,a_t^{-i})-2\sum_{t=1}^{T}\sum_{i=1}^{N}\beta_t(\delta)\sigma_t(a_t,a_{t-1})\\ &\geq \sum_{t=1}^{T}\sum_{i=1}^{N}U_t^i(a^{*,i},a_t^{-i})+\sum_{t=1}^{T}\sum_{i=1}^{N}\gamma(a_{t-1})-\sum_{t=1}^{T}\sum_{i=1}^{N}\gamma(0,a_t^{-i})\\ &\qquad -\sum_{i=1}^{N}R^i(T)-2\sum_{t=1}^{T}\sum_{i=1}^{N}\beta_t(\delta)\sigma_t(a_t,a_{t-1}).\end{aligned}$$

Let $a^{*,1:i}=(a^{*,1},\dots,a^{*,i},0,\dots,0)$ and $a^{*,1:0}=(0,\dots,0)$. Then the following holds with probability at least $1-\delta$:

$$\begin{aligned}\sum_{t=1}^{T}\gamma(a_t) &\geq \sum_{t=1}^{T}\sum_{i=1}^{N}\left[U_t^i(a^{*,i},a_t^{-i})+\gamma(a_{t-1})-\gamma(0,a_t^{-i})\right]-\sum_{i=1}^{N}R^i(T)-2\sum_{t=1}^{T}\sum_{i=1}^{N}\beta_t(\delta)\sigma_t(a_t,a_{t-1})\\ &\overset{(i)}{\geq} \sum_{t=1}^{T}\sum_{i=1}^{N}(\gamma(a^{*,i},a_t^{-i})-\gamma(0,a_t^{-i}))-\sum_{i=1}^{N}R^i(T)-2\sum_{t=1}^{T}\sum_{i=1}^{N}\beta_t(\delta)\sigma_t(a_t,a_{t-1})\\ &\overset{(ii)}{\geq} \sum_{t=1}^{T}\sum_{i=1}^{N}(\gamma(a_t+a^{*,1:i})-\gamma(a_t+a^{*,1:i-1}))-\sum_{i=1}^{N}R^i(T)-2\sum_{t=1}^{T}\sum_{i=1}^{N}\beta_t(\delta)\sigma_t(a_t,a_{t-1})\\ &\overset{(iii)}{\geq} \sum_{t=1}^{T}\sum_{i=1}^{N}(\gamma(a_t+a^{*})-\gamma(a_t))-\sum_{i=1}^{N}R^i(T)-2N\beta_T(\delta)\sum_{t=1}^{T}\sigma_t(a_t,a_{t-1}),\end{aligned}$$

where in (i) we applied Lemma 1 and similar derivations as in Inequality (3), in (ii) we used DR-submodularity (Assumption 3), and in (iii) we used a telescoping sum argument and that $\beta_t(\delta)$ is increasing in t.

Using Proposition 3, where α is the curvature, with probability at least $1-\delta$ it holds that:

$$\begin{aligned}\sum_{t=1}^{T}\gamma(a_t) &\geq \sum_{t=1}^{T}\sum_{i=1}^{N}(\gamma(a_t+a^{*})-\gamma(a_t))-\sum_{i=1}^{N}R^i(T)-2N\beta_T(\delta)\sum_{t=1}^{T}\sigma_t(a_t,a_{t-1})\\ &\geq \sum_{t=1}^{T}(1-\alpha)\gamma(a^{*})-\sum_{i=1}^{N}R^i(T)-2N\beta_T(\delta)\sum_{t=1}^{T}\sigma_t(a_t,a_{t-1}),\end{aligned}\tag{4}$$

Alternatively, using monotonicity (Assumption 3), it holds that:

$$\begin{aligned}\sum_{t=1}^{T}\gamma(a_t) &\geq \sum_{t=1}^{T}\sum_{i=1}^{N}(\gamma(a_t+a^{*})-\gamma(a_t))-\sum_{i=1}^{N}R^i(T)-2N\beta_T(\delta)\sum_{t=1}^{T}\sum_{i=1}^{N}\sigma_t(a_t,a_{t-1})\\ &\geq \sum_{t=1}^{T}(\gamma(a^{*})-\gamma(a_t))-\sum_{i=1}^{N}R^i(T)-2N\beta_T(\delta)\sum_{t=1}^{T}\sigma_t(a_t,a_{t-1}),\end{aligned}\tag{5}$$

Combining Equations (4) and (5) the following holds:

$$\frac{1}{T}\sum_{t=1}^{T}\gamma(a_t)\geq \max\left\{(1-\alpha),\frac{1}{2}\right\}\gamma(a^{*})-\sum_{i=1}^{N}\frac{R^i(T)}{T}-\frac{2N\beta_T(\delta)}{T}\sum_{t=1}^{T}\sigma_t(a_t,a_{t-1}).$$

■

B.3. Corollary for the Exp3-IX algorithm

The guarantees of Theorem 4 can be made more explicit. Let $K = \max_{i\in[N]} K^i$ and assume that the *NO-REGRET* subroutine in line 4 of Algorithm 1 is Exp3-IX for each agent. Then, we can derive the following guarantees.

Corollary 6 *Let Assumption 1, 2, and 3 hold. Then, running DO-PL (Algorithm 1) with Exp3-IX using $\eta_t = 2\xi_t = \sqrt{\log K/(Kt)}$ ensures that with probability at least $1-\delta$ the following holds:*

$$\frac{1}{T}\sum_{t=1}^{T}\gamma(a_t) \geq \max\left\{(1-\alpha), \frac{1}{2}\right\}\mathrm{OPT} - \mathcal{O}\left(N\sqrt{\frac{K\log(NK/\delta)}{T}}\right) - \mathcal{O}\left(N\sqrt{\frac{d\log T}{T}(\kappa\log(2/\delta) + d\log(T/d))}\right).$$

This is a result of substituting the high-probability regret bound of Exp3-IX [30] into Theorem 4 and upper-bounding the sum $\sum_{t=1}^{T}\sigma_t(a_t, a_{t-1})$ via Lemma 5 stated in Appendix B.1.

Proof By Theorem 4, with probability at least $(1-\frac{\delta}{2})$ the following holds:

$$\frac{1}{T}\sum_{t=1}^{T}\gamma(a_t) \geq \max\left\{(1-\alpha), \frac{1}{2}\right\}\gamma(a^*) - \underbrace{\sum_{i=1}^{N}\frac{R^i(T)}{T}}_{(I)} - \frac{2N\beta_T(\delta/2)}{T}\underbrace{\sum_{t=1}^{T}\sigma_t(a_t, a_{t-1})}_{(II)}.$$

Term (I)**:** Let $K = \max_{i\in[N]} K^i$ and assume that *NO-REGRET* in line 4 of Algorithm 1 is Exp3-IX for each agent. Then, using a union bound followed by Neu [30, Theorem 1], it holds with probability at least $(1-\frac{\delta}{2})$ that $R^i(T) \leq 2\sqrt{2TK\log(2NK/\delta)}$ for each agent [5].

Term (II)**:** Recall that $\sigma_t(a,\tilde{a}) = \|\phi(a) - \phi(\tilde{a})\|_{V_t^{-1}}$ and note that by Assumption 2 $\|\phi(a) - \phi(\tilde{a})\| \leq 2L$. Thus, applying Lemma 5 and Cauchy-Schwarz inequality it holds that:

$$\sum_{t=1}^{T}\|\phi(a_t) - \phi(a_{t-1})\|_{V_t^{-1}} \leq \sqrt{T\left(2d\log\left(1+\frac{4TL^2}{d\lambda}\right) + \frac{12dL^2}{\log(2)\lambda}\log\left(1+\frac{4L^2}{\log(2)\lambda}\right)\right)}.$$

Combining Term (I) and Term (II), and applying the union bound, with probability at least $(1-\delta)$, the following holds:

$$\frac{1}{T}\sum_{t=1}^{T}\gamma(a_t) \geq \max\left\{(1-\alpha), \frac{1}{2}\right\}\gamma(a^*) - 2N\sqrt{\frac{2K\log(2NK/\delta)}{T}} - \frac{2N\beta_T(\delta/2)}{T}\sqrt{T\left(2d\log\left(1+\frac{4TL^2}{d\lambda}\right) + \frac{12dL^2}{\log(2)\lambda}\log\left(1+\frac{4L^2}{\log(2)\lambda}\right)\right)},$$

with $\beta_T(\delta) = \mathcal{O}\left(\sqrt{\kappa\left(\log\left(\frac{1}{\delta}\right) + d\log\left(\frac{T}{d}\right)\right)}\right)$. ■

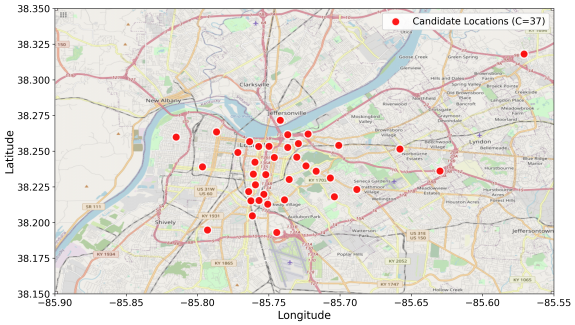

Figure 2: Candidate drop-off locations on the Louisville map. The original 50 points from a k-means algorithm applied on trip departure data are reduced to 37 points after proximity filtering.

Appendix C. Experiments

Similar to Sessa et al. [38], we empirically validate our algorithm on the problem of rebalancing a dockless shared mobility system (SMS). More specifically, for the relocation of bikes in the SMS of the city of Louisville, KY. We set $d = 4$ and aim to optimize a trade-off of the following features: number of served trips, total trip distance of the bikes, average walking distance to the bikes, and ratio of unmet demand over total demand.

Specifically, we assume the central authority operates a fleet of $N = 5$ trucks. We task each truck i to relocate $K^i = 8$ bikes to one of $C = 37$ candidate locations. The individual action set of each truck i is defined as:

$$\mathcal{A}^i := \{a^i \in \mathbb{Z}_+^C \mid \exists l \in [C] \; s.t. \; (a^i)_l = K^i \wedge (a^i)_k = 0 \; \forall k \neq l\},$$

such that $\mathcal{A} = \mathcal{A}^1 \times \cdots \times \mathcal{A}^N$ represents the set of all valid allocations.

Dataset We use a dataset from the Louisville Metro Open Data initiative. Originally available at https://data.louisvilleky.gov, the data set was retrieved from the GitHub repository of Liu et al. [25] (see Acknowledgments). For a more representative simulation, throughout the experiments, we limit historical data to the year 2019 and to trips of duration less than 60 minutes.

Defining the candidate drop-off locations We set our C candidate drop-off points to represent common departure locations for trips, while still ensuring the points are further apart, so as they can be viewed as hubs or strategic regions representatives. In order to do so, we first apply k-means clustering with $K = 50$ on the starting trips' coordinates, before filtering the centroids by only keeping those with a minimum separation of 0.5km. This results in $C = 37$ locations. A visualization of the candidate drop-off locations for our experiment in the city of Louisville can be found in Figure 2.

Features and Normalization Our welfare function $\gamma(a) = \langle \theta^*, \phi(a) \rangle$ corresponds to a trade-off between the four features $\phi(a) = \{\phi(a)_i\}_{1 \leq i \leq 4}$ described below for any given allocation a.

- Met demand: we compute the met demand as the number of trips served during the day.

- Total trip distance of the bikes: we sum the trip distance realized by all the bikes during the day. In practical scenarios, this might be strongly linked to the revenue generated for the day.
- Average walking distance: for served trips, we compute the average walking distance that a user had to perform to start their trip. This is a metric we typically wish to minimize.
- Unmet demand ratio: the ratio of the number of unmet trips over the total number of trips for the day. This is also a metric we wish to minimize.

We further normalize each feature to have a truncated standard normal distribution in [0, 1], so that a random strategy would have a feature expectation of 0.5 for a step chosen uniformly at random.

Notes on Centralized Optimization In general, for N players each with A possible actions, the joint action space is of size A^N. In our application, each truck has to choose one of 37 candidate locations independently, resulting in a total of $37^5 \approx 6.93 \times 10^7$ possible allocations. This makes central optimization intractable and motivates our need for distributed optimization. Representative values for multiple values of C and N are shown in Table 1.

Table 1: Values of C^N for different action set size C and number of players N. In scientific notation with 2 decimal places.

Base (C)	$N = 2$	$N = 5$	$N = 10$
2	4.00×10^0	3.20×10^1	1.02×10^3
4	1.60×10^1	1.02×10^3	1.05×10^6
8	6.40×10^1	3.28×10^4	1.07×10^9
16	2.56×10^2	1.05×10^6	1.10×10^{12}

Simulator Setup To simulate this scenario, we leverage a dataset from the Louisville Metro Open Data, comprising records of e-scooter and bikesharing trips in a free-floating system in the city of Louisville.[7] These trips are used to estimate the users' demand in an underserved scenario with only 40 vehicles. At the beginning of each day t, the vehicles are placed at the hub locations captured by a_t, where a_t^i represents the unique drop-off hub selected by truck i. During the day, we meet demand in chronological order whenever possible, with the simplifying assumption that no trips overlap. That is, for each trip, if a vehicle is accessible at a maximal walking distance estimate of 0.5km, we consider the demand to be met and relocate the vehicle to the end trip location. If the trip cannot be realized, the vehicle stays in place, and the demand is unmet.

This simulator is used to compute the metrics $\phi(a_t)$ for any given joint allocation a_t.

Learning Optimal Allocations In practice, it is unrealistic that we can measure $\phi(\overline{a})$ for some allocation $\overline{a} \in \mathcal{A}$ that is not played by the algorithm. We therefore implement DO-PL with the

7. Originally available at https://data.louisvilleky.gov, the data set was retrieved from the GitHub repository of Liu et al. [25].

Exp3-IX no-regret learning algorithm [30] to learn from features corresponding to executed allocations. For each experiment, the reference $\theta^* \in \Delta_3$ is sampled from a 4-dimensional flat Dirichlet distribution and fixed throughout all the $T = 400$ environment steps.[8] The specific distribution of θ^* is not important, but constraining θ^* to a probability simplex via the Dirichlet distribution allows for better interpretation of the weights as trade-off parameters on the metrics.

When applying DO-PL, knowledge of θ^* is only used for preference sampling that the algorithm uses to learn.

Experiments and results We compare the performance of DO-PL against a random baseline, taking actions uniformly at random, and the D-SubUCB algorithm of Sessa et al. [38]. We note that while we optimize from preference feedback, D-SubUCB directly observes the realized welfare values $\gamma(a_t)$ for each executed action a_t.

1. Average welfare over episodes

For completeness, we restate below the performance result present in the main content.

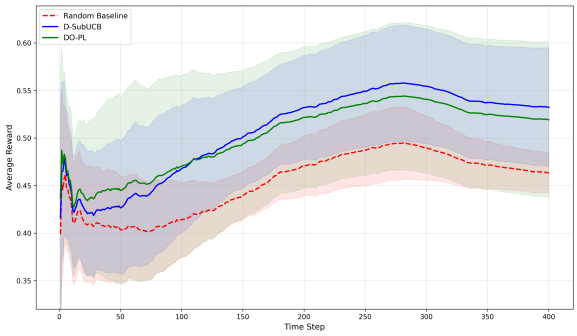

Figure 3: Average reward for the three online algorithms (D-SubUCB, DO-PL, Random Baseline) over $T = 400$ steps, averaged over 100 experiments. For each experiment, we sample $\theta^* \sim \text{Dirichlet}(4)$ and run the algorithms in parallel using the same source of randomness.

Figure 3 shows that DO-PL significantly outperforms the random baseline over the entire horizon. It does so, while slightly underperforming D-SubUCB that directly learns from the realized welfare values.

2. Trading-off the different features

Maximizing a linear welfare function γ is equivalent to maximizing a trade-off of objectives described by the features ϕ and their corresponding weights θ^*. The experiment above focused on the performance of the three algorithms (D-SubUCB, DO-PL, Random baseline) averaged over various θ^* values sampled at random.

8. For any $k \geq 1$, $\Delta_k \subseteq \mathbb{R}^{k+1}$ denotes the k-dimensional simplex, as the set of probability vectors in $\mathbb{R}^{k+1}$.

To further validate our algorithm, we ran experiments with a fixed θ^*[9], and compute the averaged metrics over $T = 400$ timesteps and 3200 runs for each algorithm. Results of this experiment are shown in Figure 4.

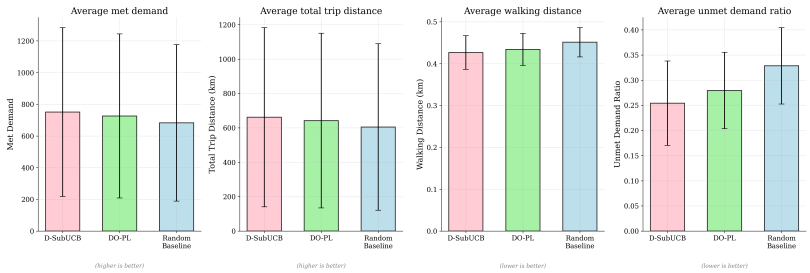

Figure 4: Average metrics over $T = 400$ rounds for the three online algorithms (D-SubUCB, DO-PL, Random Baseline). This is for a fixed $\theta^* \approx (0.25, 0.33, 0.33, 0.08)$, with averages computed over 3200 runs.

When θ^* is fixed, our algorithm improves on the four metrics presented above compared to the baseline. Indeed, in Figure 4 we show that DO-PL does better than the random baseline on the sub-objectives defined by the features.

3. Comparison with OPT

We then compare the performance of our algorithm by comparing it to OPT for various values of the horizon T. Specifically, for each T, we tune the hyperparameters of our algorithm and compute the average rewards over 600 runs.

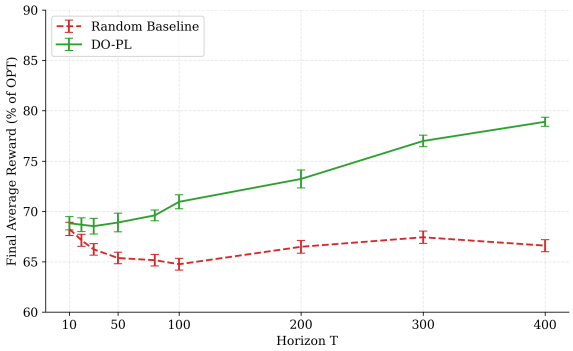

Figure 5: Final average reward as a percentage of OPT for different values of T. Computed over 600 runs for DO-PL and the Random Baseline. OPT is computed by simulations of an exhaustive search on the joint allocations.

9. θ^* is fixed, but the algorithms remain independent of its value. In summary, D-SubUCB depends on the rewards directly, DO-PL only depends on the observed features, and the random baseline is independent of the environment parameters.

Figure 5 is coherent with the regret bounds of our main result Theorem 4.

4. Impact of temperature

Additionally, we naturally extend our algorithm and the preference feedback computation to depend on a temperature parameter τ, effectively multiplying the logits of the Bradley-Terry model. The experiments reported in the main text are performed by setting $\tau = 1.0$. However, we provide in Figure 6 an analysis of the impact of this parameter on the performance of DO-PL, when all the other hyperparameters are tuned for each temperature. We see that when the preference feedbacks become more reliable, i.e. when the temperature is increased, the average performance of our algorithm is improved. With a low temperature of $\tau = 0.1$, DO-PL is comparable with the random baseline, as the learning signal is of low quality (very noisy feedbacks). With a high temperature of $\tau = 5.0$, DO-PL is comparable to D-SubUCB [38] in terms of performance, as seen in Figure 1.

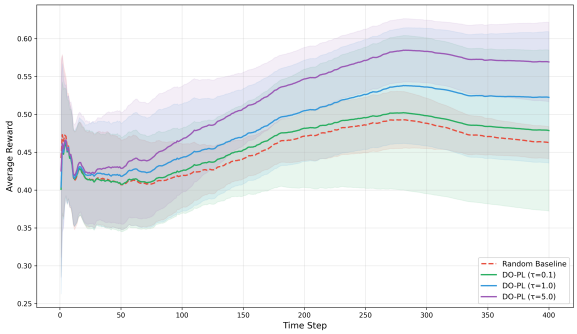

Figure 6: For a fixed learning rate, we show the impact of varying the temperature parameter τ. Performance is highly sensitive to τ.

Hyperparameter tuning For each algorithm, we perform independent hyper-parameter tuning by taking the set of hyperparameters maximizing the average rewards over $T = 400$ rounds, averaging the results over 50 independent runs. In particular, we tune the following hyperparameters:

- DO-PL: a fixed β value, the learning rate for the Exp3-IX updates [5]
- D-SubUCB: a fixed β value, the learning rate for the Multiplicative weights update (MWU, see Freund and Schapire [15], Sessa et al. [38])

For the Gaussian process model of D-SubUCB [38], we randomly generated an offline dataset of 200 days used for tuning.