Online Submodular Resource Allocation with Applications to Rebalancing Shared Mobility Systems

Pier Giuseppe Sessa [1] **Ilija Bogunovic** [1] **Andreas Krause** [1] **Maryam Kamgarpour** [1]

Abstract

Motivated by applications in shared mobility, we address the problem of allocating a group of agents to a set of resources to maximize a cumulative welfare objective. We model the welfare obtainable from each resource as a monotone *DR-submodular* function which is a-priori *unknown* and can only be learned by observing the welfare of selected allocations. Moreover, these functions can depend on time-varying contextual information. We propose a distributed scheme to maximize the cumulative welfare by *designing* a *repeated game* among the agents, who learn to act via regret minimization. We propose two design choices for the game rewards based on upper confidence bounds built around the unknown welfare functions. We analyze them theoretically, bounding the gap between the cumulative welfare of the game and the highest cumulative welfare obtainable in hindsight. Finally, we evaluate our approach in a realistic case study of rebalancing a shared mobility system (i.e., positioning vehicles in strategic areas). From observed trip data, our algorithm gradually learns the users' demand pattern and improves the overall system operation.

1. Introduction

A number of important real-world problems consist of repeatedly allocating agents to resources to maximize a cumulative welfare objective. Examples include positioning sensors to maximize the probability of detecting an event (Krause & Guestrin, 2007), or allocating bandwidth to nodes of a wireless network (Stanczak et al., 2006). These problems have far-reaching applications in several areas of science and engineering (Katoh & Ibaraki, 1998).

In certain applications, however, the welfare objective function (i.e., the quality measure of a given allocation) is a-priori *unknown* and can only be learned *online* by observing the outcome of proposed allocations. Moreover, outcomes can depend on external varying *contextual* factors, e.g., time, weather, etc.

A concrete example scenario, which we address in Section 6, is the problem of *rebalancing* a Shared Mobility System (SMS), such as bike or scooter sharing. Here the goal is to strategically reposition vehicles in a city (typically overnight by using transportation trucks) to increase the vehicles' availability and minimize the users' dissatisfaction level. SMSs have experienced tremendous growth over the past decade, offering a compelling alternative to classic transportation systems. Their potential benefits are numerous, including sustainability, increase in efficiency, and reduction of costs, among many others (Laporte et al., 2018). Effective rebalancing of such systems, however, is key to their success. In these application, the exact demand for vehicles is unknown ahead of time and depends on external factors such as weather.

Motivated by this important application, we consider the problem of *online resource allocation with unknown welfare functions*. We propose a distributed approach that provably attains near-optimal solutions to this problem, and we demonstrate the performance in a realistic case study on data from an SMS in Louisville (KY, US).

1.1. Related work

Distributed resource allocation. Resource allocation problems are typically addressed by distributed protocols, where each agent is assigned to one or more resources based on local computations. The general game-theoretic framework of Marden & Wierman (2013) proposes to *design a game* between the agents and retrieve allocations by computing the resulting game equilibria. In the case of *submodular* welfare functions (which is a typical assumption in these problems), games can be designed according to Vetta (2002) so that such equilibria attain at least a 0.5-approximation to the optimal achievable welfare. These guarantees hold also in the online setting where agents act via regret minimization (Blum et al., 2008), and with continuous strategy sets (Sessa et al., 2019a). Typical game design choices,

[1]ETH Zürich, Switzerland. Correspondence to: Pier Giuseppe Sessa <sessap@ethz.ch>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

however, assume the welfare functions are fully known and can be evaluated at different game actions. We build on the framework of Marden & Wierman (2013), with the important difference that we deal with *unknown* welfare functions, which may also depend on time-varying contextual information. This leads to new trade-offs and challenges in designing suitable games.

Bayesian optimization. An independent body of literature focuses on optimizing unknown functions from sequential noisy evaluations. Several algorithms using Bayesian non-parametric models have been developed over the years (e.g., Mockus, 1989; Srinivas et al., 2010; Chowdhury & Gopalan, 2017; Krause & Ong, 2011; Bogunovic et al., 2018; Sessa et al., 2019a), under different assumptions. They use Gaussian process (Rasmussen & Williams, 2005) regression techniques to build a confidence interval around the unknown objective function, and can implicitly balance exploration (select points with high uncertainty) and exploitation (select high-rewarding points). However, these algorithms are intractable in our resource allocation setup, since the set of possible allocations to be considered is *exponential* in the number of agents. Instead, our distributed approach employs the aforementioned techniques to maintain an *upper confidence bound* on the welfare functions and uses these bounds to compute game rewards for the agents.

Submodular optimization. The problem considered in Section 2 can also be abstracted as an online maximization of a sequence of monotone submodular functions, for which *greedy* algorithms can converge to $(1-1/e)$-approximation guarantees under different constraint sets (Golovin et al., 2014; Zhang et al., 2019). However, they require multiple evaluations of these functions to compute marginal contributions (or their gradients in continuous domains). Chen et al. (2017), instead, consider unknown functions and assume noisy observations of the marginal contributions. Compared to these works, we assume observing *only* the welfare of the selected allocations (i.e., the so-called bandit feedback). Zhang et al. (2019) consider online submodular *bandit* optimization but with a significantly slower convergence due to the use of high-variance gradient estimators. Instead, in this work we impose kernel-based regularity assumptions which allow us to learn the welfare functions online from past observed data.

Truck-based rebalancing of SMSs. Several truck-based rebalancing strategies for SMSs have been proposed in the literature. A large body of works (e.g., Dell'Amico et al., 2014, and references therein), employ mathematical programming techniques to find optimal routes for the trucks, given a pre-specified vehicles' positioning plan. These results are complementary to our work in that we focus on optimizing the latter. The line of works initiated by Ghosh et al. (2015) addresses dynamic allocations of vehicles to stations, proposing a mixed-integer robust optimization framework that considers a finite set of possible demand scenarios. Their problem size increases with the number of trucks, prediction horizon, and possible scenarios, and is solved using Lagrangian duality techniques. Compared to these works, albeit we only focus on static (i.e., overnight) rebalancing, our approach is distributed and uses observed demand data to learn about users' demand. Jian et al. (2016) find strategic vehicle allocations using simulation-based optimization heuristics, while Bhatia et al. (2019) propose a deep reinforcement learning approach. Both these methods, however, require access to a reliable simulator and a large number of evaluations. Other works, use historical data to fit suitable models (e.g., station-based Poisson processes, Freund et al. (2020), Neural Networks, Lin et al. (2018), Random Forests, Yang et al. (2016)) to predict users' demand and use them, in a separate step, for repositioning. Compared to these methods, our approach gradually learns the users' demand patterns *online*. The use of upper confidence bound functions allows to implicitly trade-off between attempting new rebalancing strategies (exploration) and focusing on high-rewarding allocations according to the data observed so far (exploitation). As we show in our experimental Section 6, this allows us to effectively learn the users' demand patterns and produce efficient allocations after a few days of operation.

1.2. Contributions.

We address the problem of online resource allocation with unknown and context-dependent welfare.

- We propose a distributed algorithm, D-SUBUCB (Distributed Submodular Upper Confidence Bound) which simulates a repeated game among the agents, who act via regret minimization, and computes game rewards based on upper confidence bound techniques.
- We theoretically analyze two game design choices, bounding the gap between the game cumulative welfare and the welfare obtainable in hindsight by a best fixed allocation or, in case contexts are observed at decision time, by the best *policy* mapping contexts to allocations.
- We formulate the problem of rebalancing a Shared Mobility System according to our model and showcase the performance of our approach in rebalancing the shared system of Louisville, KY, based on historical trip data.

1.3. Notation.

We denote with $\mathbf{e}_i$, $\mathbf{0}$, and $\mathbf{1}$, the i^{th} unit vector, null vector, and the all-one vector of appropriate dimension, respectively, while $I_n \in \mathbb{R}^{n\times n}$ is the identity matrix. Given $x, y \in \mathbb{R}^d$, we let $x[i]$ be the i^{th} coordinate and $x \leq y$ be a coordinate-wise inequality. We also define $[n] := \{1, \dots, n\}$.

2. Problem formulation

We consider the sequential decision-making problem of allocating N agents among R resources repeatedly over time. We let $\mathcal{X}^i \subset [0, x_{\max}]^R$ be the decision set for agent i and $\boldsymbol{\mathcal{X}} := \mathcal{X}^1 \times \cdots \times \mathcal{X}^N$ be the whole strategy space. In our SMS application, e.g., N is the number of trucks used for rebalancing, each with a capacity of $x_{\max}$ vehicles which can be positioned over R candidate regions. Moreover, at each round t, we allow the decision-making problem to be influenced by a (potentially different) *context* vector $z_t \in \mathcal{Z}$ (e.g., time, weather, etc.). The quality of a given allocation (i.e., rebalancing strategy in a SMS) $\mathbf{x}_t \in \boldsymbol{\mathcal{X}}$ is measured by the welfare function $\gamma : \boldsymbol{\mathcal{X}} \times \mathcal{Z} \to \mathbb{R}_+$ which we define as an additive function:

$$\gamma(\mathbf{x}_t, z_t) := \sum_{r=1}^{R} \gamma^r(\mathbf{x}_t, z_t)\,, \tag{1}$$

where each $\gamma^r(\cdot)$ measures the welfare gained from resource r. Note that (1) is without loss of generality in that γ is generally non-separable over the resources.[1] That is, each welfare function γ^r does not depend exclusively on the variables $\{x_t^i[r], i \in [N]\}$ but on the whole allocation vector $\mathbf{x}_t$. In the SMS application, $\gamma(\mathbf{x}_t, z_t)$ might quantify the total number of trips completed in day t (i.e., number of users who could successfully utilize the system), where each $\gamma_r(\cdot)$ measures the number of trips started from region r. The welfare is non-separable as the number of trips starting from region r depends also on the vehicles $x_t^i[j], j \neq r$, since users could bring more vehicles from region j to region r during morning hours. Our goal is to allocate agents to resources to maximize the cumulative welfare $\sum_t \gamma(\mathbf{x}_t, z_t)$.

Many resource allocation problems, such as those in the SMS domain satisfy two important properties: Monotonicity and submodularity, a natural notion of diminishing returns. In particular, we assume each $\gamma^r(\cdot, z)$ is a *monotone* (i.e., for all $\mathbf{x}_1 \leq \mathbf{x}_2 \in \boldsymbol{\mathcal{X}}$, $\gamma(\mathbf{x}_1, z) \leq \gamma^r(\mathbf{x}_2, z)$) and *DR-submodular* function for each $z \in \mathcal{Z}$, as defined below. Without loss of generality we also assume $\gamma^r(\mathbf{0}, z) = 0, \forall z$.[2]

Def 1 (DR-Submodularity, (Bian et al., 2017))**.** *A function* $f : \boldsymbol{\mathcal{D}} \subseteq \mathbb{R}^d \to \mathbb{R}$ *is* DR-submodular *if, for all* $\mathbf{x} \leq \mathbf{y} \in \boldsymbol{\mathcal{D}}$, $\forall i \in [d]$, $\forall k \geq 0$ *such that* $(\mathbf{x} + k\mathbf{e}_i)$ *and* $(\mathbf{y} + k\mathbf{e}_i) \in \boldsymbol{\mathcal{D}}$,

$$f(\mathbf{x} + k\mathbf{e}_i) - f(\mathbf{x}) \geq f(\mathbf{y} + k\mathbf{e}_i) - f(\mathbf{y})\,.$$

When $\gamma^r(\cdot, z)$ is twice-differentiable, it is DR-submodular whenever all entries of its Hessian are non-positive. Moreover, in case of binary domains $\boldsymbol{\mathcal{D}} = \{0, 1\}^d$, Def 1 coincides with submodularity of set functions (Bach, 2019). Note that these assumptions are verified in our SMS application since increasing the number of vehicles leads to a higher number of trips and lower marginal returns (we provide more details in Section 6). Finally, note that the above setup includes also several well-studied problems such as sensor placement, vehicle target assignment, and graph coloring (see Marden & Wierman, 2013, and references therein).

Compared to previous works, here we focus on the challenging setting in which the functions $\{\gamma^r(\cdot), r \in [R]\}$ are a-priori *unknown*, and we can only learn them online by selecting allocations $\mathbf{x}_t$, and observing the noisy rewards:

$$r_t^r = \gamma^r(\mathbf{x}_t, z_t) + \xi_t^r\,, \quad r = 1, \ldots, R \tag{2}$$

where ξ_t^r is σ-sub-Gaussian noise. This is the case in SMSs, where $\gamma(\cdot)$ does not have a closed-form expression as it depends on the complex users' demand patterns, and we can only observe the outcome of selected rebalancing strategies.

Performance benchmark. We make no assumption on how the contexts z_t's are generated [3] and, after T rounds, we consider the natural benchmark:

$$\mathrm{OPT} = \max_{\mathbf{x} \in \boldsymbol{\mathcal{X}}} \sum_{t=1}^{T} \gamma(\mathbf{x}, z_t)\,, \tag{3}$$

i.e., the best cumulative reward obtainable by a single strategy if the sequence of contexts and the welfare functions were *known* ahead of time. In Section 5, assuming that context z_t can be observed *before* choosing $\mathbf{x}_t$, we consider the stronger benchmark of finding the best *policies* mapping contexts to allocations.

Problem (3) can be seen as an instance of online submodular maximization, which is in general NP-hard (Golovin et al., 2014). Moreover, when only bandit feedback (2) is available, existing algorithms (Zhang et al., 2019) converge to a $(1 - 1/e)^{-1}$ approximation with a slow rate of $\mathcal{O}(T^{8/9})$. In this work, we take a different approach and make a smoothness assumption on the welfare functions. Namely, we assume each γ^r has a bounded (and small) norm $||\gamma^r||_{k^r} \leq B$ in a Reproducing Kernel Hilbert Space (RKHS) associated to a kernel function $k^r : (\boldsymbol{\mathcal{X}} \times \mathcal{Z}) \times (\boldsymbol{\mathcal{X}} \times \mathcal{Z}) \to \mathbb{R}_+$. Typical kernel choices are polynomial, squared-exponential, and Màtern kernels (see, e.g., Frazier, 2018, and references therein). This is a non-parametric assumption widely used in Bayesian optimization which, as outlined later, allows us to use the observed data to efficiently learn about unseen outcomes.

Several algorithms can optimize unknown functions subject to the discussed smoothness and feedback model. For instance, GP-MW (Sessa et al., 2019a) and GP-UCB (Srinivas et al., 2010) can provably converge to OPT under adversarial or known contexts' sequences, respectively. However, in our resource allocation setup, their computational complexity scales *exponentially* with the number of agents N

[1] This is different from, e.g., Marden & Wierman (2013); Paccagnan et al. (2020) who consider separable welfare functions.

[2] Otherwise one could treat these terms as constant offsets.

[3] In fact, they can be chosen by an adaptive adversary (Cesa-Bianchi & Lugosi, 2006) as a function of the data up to $t - 1$.

Algorithm 1 Example of NO-REGRET algorithm class, with update rule of MWU (Freund & Schapire, 1997)

Require: Set $\mathcal{X}$ with $|\mathcal{X}| = K$, learning rates $\{\eta_t\}_{t=1}^T$.
  $\mathbf{w} = 1/_K \cdot [1, \dots, 1] \in \mathbb{R}^K$. `// initialize weights`
  Def `play_action()`:
    $\mathbf{p} = \mathbf{w} \cdot 1/_{\sum_{i=1}^N \mathbf{w}[i]}$ `// mixed strategy`
    $x \sim \mathbf{p}$ `// sample action`
  Return x
  Def `update(`$f(\cdot)$`)`:
    $\mathbf{f} = [f(x)]_{x \in \mathcal{X}} \in \mathbb{R}^K$ `// rewards vector`
    $\mathbf{w} = \mathbf{w} \cdot \exp(\eta_t \mathbf{f})$ `// MWU update rule`
  Return

(as they require to iterate over the set $\boldsymbol{\mathcal{X}}$ of possible allocations) and hence they become intractable even for small problem instances. Instead, the approach proposed in this work builds on the *distributed* game-theoretic framework of Marden & Wierman (2013), with the main difference of dealing with unknown and context-dependent welfare functions. Moreover, compared to Marden & Wierman (2013), we consider a more general objective which is non-separable. This leads to new trade-offs and challenges that we address next.

3. Proposed approach

Our approach utilizes two main interconnected algorithmic components. The first component consists of computing allocations by designing and simulating a *repeated game* among the agents, while the second one relies on RKHS regression to build suitable confidence bounds around the unknown functions γ^r. They are presented in the next two subsections. Then, in Section 4 we propose and analyze two concrete game design choices.

3.1. Designing game dynamics

To maximize the cumulative welfare $\sum_t \gamma(\mathbf{x}_t, z_t)$, we exploit the decoupled constraint structure by simulating a repeated game among the N agents, or players in the game. At every round t, each player i selects action $x_t^i \in \mathcal{X}^i$ based on its past observations, as outlined below. Then, we build allocation $\mathbf{x}_t = [x_t^1, \dots, x_t^N]$ as the joint vector of actions played. We orchestrate the coordination of the players via *designing* suitable reward functions, which the players learn to selfishly optimize. By careful design of these rewards, we aim to maximize the social welfare. We discuss specific choices in Section 4. At each time t, we denote the reward function of each player i by $f_t^i : \mathcal{X}^i \to \mathbb{R}$. Concerning the players' behavior, we let each player act and update its strategy according to a *no-regret* algorithm (Cesa-Bianchi & Lugosi, 2006). Given a sequence of reward functions $f_1^i, \dots, f_T^i$, the regret of player i is defined as

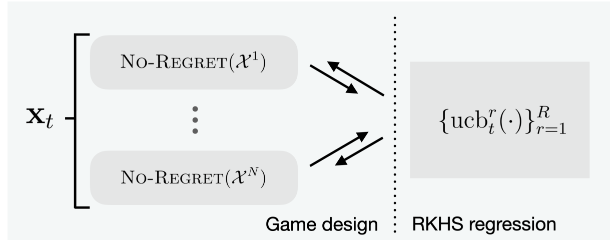


Figure 1. A sketch of the D-SUBUCB algorithm

Algorithm 2 The D-SUBUCB algorithm

Require: Set $\boldsymbol{\mathcal{X}} = \prod_{i=1}^N \mathcal{X}^i$, kernels $\{k^r\}_{r=1}^R$, $\{\beta_t^r\}_{t=1}^T$.
  `/* Initialize learning algorithms */`
  Algo^i $\leftarrow$ NO-REGRET($\mathcal{X}^i$), $\quad i = 1, \dots, N$
  for $t = 1, \dots, T$ **do**
    Nature chooses context z_t
    `/* In parallel, sample actions */`
    $x_t^i \leftarrow$ Algo^i`.play_action()`, $\quad i = 1, \dots, N$
    Select strategy $\mathbf{x}_t = [x_t^1, \dots, x_t^N]$
    for $r = 1, \dots, R$ **do**
      Observe noisy reward $r_t^r = \gamma^r(\mathbf{x}_t, z_t) + \xi_t^r$
      Update $\mathrm{ucb}_t^r(\cdot)$ according to (4) and (5).
    end for
    `/* In parallel, provide feedback to algs. */`
    Compute game rewards $\{f_t^i(\cdot)\}_{i=1}^N$ via (6) or (8).
    Algo^i`.update(`$f_t^i(\cdot)$`)`, $i = 1, \dots, N$
  end for

Def 2 (Regret of player i). *The regret of player i after T game rounds is*

$$R^i(T) = \max_{x \in \mathcal{X}^i} \sum_{t=1}^T f_t^i(x) - \sum_{t=1}^T f_t^i(x_t^i)\,.$$

Under different game assumptions, several no-regret algorithms exist ensuring that $R^i(T)/T \to 0$ as $T \to \infty$ (in expectation, or with high probability). As an example of such algorithms, MWU (Freund & Schapire, 1997) presented in Algorithm 1, can be used in case $\mathcal{X}^i$ is finite.

Note that the proposed approach is parallelizable across the N players, and therefore its parallel computational complexity does not scale with N (even if players' strategies are updated sequentially, it scales linearly on N as opposed to considering the exponential action space $\boldsymbol{\mathcal{X}}$). Nevertheless, we are left with the important task of designing suitable players' reward functions that can steer the game to high welfare. The proposed design choices rely on the following RKHS regression techniques.

3.2. RKHS regression

At every round t, and for each resource r, standard kernel ridge regression on the past observed data $\{\mathbf{x}_\tau, z_\tau, r_\tau^r\}_{\tau=1}^t$

allows us to compute posterior mean and variance estimates of the unknown welfare function $\gamma_r(\cdot)$, respectively as:

$$\begin{aligned}\mu_t^r(\mathbf{x}, z) &= k_t(\mathbf{x}, z)^T \big(K_t + \lambda I_t\big)^{-1} \boldsymbol{r}_t \qquad (4)\\ \sigma_t^r(\mathbf{x}, z)^2 &= k^r(\mathbf{x}, z, \mathbf{x}, z) \\ &\quad - k_t(\mathbf{x}, z)^T \big(K_t + \lambda I_t\big)^{-1} k_t(\mathbf{x}, z)\,,\end{aligned}$$

where $k_t(\mathbf{x}, z) = [k^r(\mathbf{x}, z, \mathbf{x}_1, z_1), \dots, k^r(\mathbf{x}, z, \mathbf{x}_T, z_T)]^T$, $[K_t]_{i,j} = k^r(\mathbf{x}_i, z_i, \mathbf{x}_j, z_j)$ is the kernel matrix, $\lambda > 0$ is a regularization parameter and $\boldsymbol{r}_t = [r_1^r, \dots, r_t^r]^T$ is the vector of noisy observations. Moreover, these can be used to construct the upper confidence bound function:

$$\mathrm{ucb}_t^r(\mathbf{x}, z) := \mu_t^r(\mathbf{x}, z) + \beta_t^r \sigma_t^r(\mathbf{x}, z)\,, \qquad (5)$$

where β_t^r is a tunable confidence parameter. A main result from Srinivas et al. (2010); Abbasi-Yadkori (2013) (see Lemma 1 in Appendix A.1) shows that under our regularity assumptions, β_t^r can be set such that, with high probability, $\mathrm{ucb}_t^r(\mathbf{x}, z) \geq \gamma^r(\mathbf{x}, z)$ for all $\mathbf{x}, z$, and $t \geq 1$.

We next utilize the upper confidence bound functions computed in (5) to design suitable reward functions f_t^i's for the players. Our overall approach is summarized in the proposed (meta) algorithm D-SUBUCB (Distributed Submodular Upper Confidence Bound), sketched in Figure 1 and outlined in Algorithm 2.

4. Design choices and guarantees

4.1. Total Welfare (TW) design

Under Total Welfare (TW) design, the rewards for each player i at round t are computed as:

$$f_t^i(x) \overset{\text{TW}}{:=} \sum_{r=1}^{R} \mathrm{ucb}_t^r(x, x_t^{-i}, z_t), \quad x \in \mathcal{X}^i, \qquad (6)$$

i.e., as an aggregate upper confidence bound on the total game welfare, under opponents' actions x_t^{-i}. The idea behind this design choice can intuitively be explained as follows. As the game proceeds and more data are available, the ucb_t^r's functions converge to the true welfare functions. At the same time, as shown by Vetta (2002) and Sessa et al. (2019a), the DR-submodularity property (Def 1) ensures that the players' rewards $f_t^i(\cdot)$ are 'aligned' with the total welfare $\gamma(\cdot, z)$ for each z. Therefore, by minimizing their regret, the players (and hence the allocations computed by D-SUBUCB) obtain high welfare and, as more precisely stated in the next theorem, achieve provable approximation guarantees to (3). We relegate its proof to Appendix A.3.

The obtained guarantees depend on the notions of *average* and *worst-case game curvature* defined below.

Def 3 (Game curvatures). *Consider a sequence of contexts* $z_1, \dots, z_T$. *We define* average *and* worst-case game curvature*, as:*

$$c_{avg}(\{z_t\}_{t=1}^T) = 1 - \inf_i \frac{\sum_{t=1}^T \big[\nabla\gamma(2\mathbf{x}_{max}, z_t)\big]_i}{\sum_{t=1}^T \big[\nabla\gamma(\mathbf{0}, z_t)\big]_i} \in [0,1]\,,$$

$$c_{wc}(\{z_t\}_{t=1}^T) = 1 - \inf_{t,i} \frac{\big[\nabla\gamma(2\mathbf{x}_{max}, z_t)\big]_i}{\big[\nabla\gamma(\mathbf{0}, z_t)\big]_i} \in [0,1]\,,$$

where $\mathbf{x}_{max} = x_{max}\mathbf{1}$.

The average game curvature coincides with the curvature (Sessa et al., 2019b, Definition 2) of the DR-submodular function $\gamma_{\text{avg}}(\cdot) = \sum_{t=1}^T \gamma(\cdot, z_t)/T$ which describes the time-averaged game, with respect to the set $[0, 2x_{\max}]^R$. Instead, $c_{\text{wc}}(\{z_t\}_{t=1}^T)$ quantifies the worst-case curvature over the game rounds. In Appendix A.2, we define these notions in the more general case where γ is non-differentiable. Both notions measure how close $\gamma(\cdot, z)$ is from being linear, in which case $c_{\text{avg}}(\{z_t\}_{t=1}^T) = c_{\text{wc}}(\{z_t\}_{t=1}^T) = 0$ and the optimization goal (3) becomes separable over the N agents. In general, it holds $0 \leq c_{\text{avg}}(\{z_t\}_{t=1}^T) \leq c_{\text{wc}}(\{z_t\}_{t=1}^T) \leq 1$ (see Appendix A.2, Lemma 3).

Thm 1. *Consider the setup of Section 2. When* D-SUBUCB *is run with TW design (rule* (6)*) and* β_t^r*'s are set according to Lemma 1 (Appendix A.1), with high probability,*

$$\sum_{t=1}^T \gamma(\mathbf{x}_t, z_t) \geq \alpha \cdot \mathrm{OPT} - N \sum_{t=1}^T \sum_{r=1}^R 2\beta_t^r \sigma_t^r(\mathbf{x}_t, z_t) - \sum_{i=1}^N R^i(T)\,,$$

with $\alpha = \max\big\{1 - c_{avg}(\{z_t\}_{t=1}^T),\ \big(1 + c_{wc}(\{z_t\}_{t=1}^T)\big)^{-1}\big\}$.

The guarantees obtained in Thm 1 can be made more explicit by defining, for each resource r, the *maximum information gain* (Srinivas et al., 2010):

$$g_T^r := \max_{\{(\mathbf{x}_t, z_t)\}_{t=1}^T} 0.5 \log\det(I_T + K_T/\lambda). \qquad (7)$$

This sample complexity parameter quantifies the reduction in uncertainty about $\gamma^r(\cdot)$ after T noisy observations. Moreover, assume $|\mathcal{X}^i| = K$ and that MWU (Algorithm 1) is used for each player. Then, we can conclude the following.

Corollary 1. *Consider the setup of Section 2 and assume* $|\mathcal{X}^i| = K$ *for all* i*. Then, if* D-SUBUCB *is run with TW design,* $\beta_t^r = B + \sigma\lambda^{-1/2}\sqrt{2(g_t^r + \log(2/\delta))}$ *and* NO-REGRET *is* MWU *(Algorithm 1), with probability* $1 - \delta$,

$$\sum_{t=1}^T \gamma(\mathbf{x}_t, z_t) \geq \alpha \cdot \mathrm{OPT} - N \sum_{r=1}^R \mathcal{O}\big(g_T^r \sqrt{T}\big) - N \cdot \mathcal{O}\big(\sqrt{T \log K} + \sqrt{T \log(2/\delta)}\big)\,,$$

with $\alpha = \max\big\{1 - c_{avg}(\{z_t\}_{t=1}^T),\ \big(1 + c_{wc}(\{z_t\}_{t=1}^T)\big)^{-1}\big\}$.

The above guarantee is obtained from Thm 1 by substituting the well-known kernel-dependent $\mathcal{O}\big(g_T^r\sqrt{T}\big)$ bound on the sum of posterior standard deviations (see Lemma 2 in Appendix A.1), and the high probability regret bound of MWU (Freund & Schapire, 1997). Further, g_T^r can be bounded analytically for popularly used kernels, e.g., when $\boldsymbol{\mathcal{X}} \times \mathcal{Z} \subset \mathbb{R}^d$, $g_T^r \leq \mathcal{O}(\log(T)^{d+1})$ and $g_T^r \leq \mathcal{O}(d\log(T))$ for squared exponential and linear kernels, respectively (Srinivas et al., 2010) . This shows that, as $T \to \infty$, D-SubUCB approaches sublinearly an α-approximation of OPT, with $\alpha \in [0.5, 1]$. Such approximation generalizes the ones of Vetta (2002); Sessa et al. (2019b) to the case of a context-dependent welfare. It is an a-posteriori performance guarantee, as it depends on the sequence of observed contexts. Moreover, it depends on the average game sequence instead of only considering the worst-case context as done in (Sessa et al., 2020). Finally, note that $c_{\text{avg}}(\{z_t\}_{t=1}^T)$ and $c_{\text{wc}}(\{z_t\}_{t=1}^T)$ cannot be computed as $\gamma(\cdot)$ is unknown, but they could be estimated, e.g., using its posterior mean.

4.2. Anonymous game with binary strategy sets: Equal Share (ES) design

In this section we define an alternative design choice, for the special case in which the strategy spaces are binary, $\mathcal{X}^i = \{0, x_{\text{max}}\}^R$, and the game is *anonymous*. To define an anonymous game, we first introduce some helpful notation. For a given allocation $\mathbf{x} \in \boldsymbol{\mathcal{X}}$, we let $|\mathbf{x}|_r$ denote the number of players allocating a non-zero quantity in resource r, i.e., $|\mathbf{x}|_r := |\{i : x^i[r] > 0\}|$. A game is called *anonymous* if, for each resource r and any pair $\mathbf{x}_1, \mathbf{x}_2 \in \boldsymbol{\mathcal{X}}$ such that $|\mathbf{x}_1|_i = |\mathbf{x}_2|_i$ for all $i \in [R]$, $\gamma^r(\mathbf{x}_1, z) = \gamma^r(\mathbf{x}_2, z)$ for all z. Perhaps the most natural type of anonymous game is when $\gamma^r(\mathbf{x}, z) = \gamma^r(\sum_{i=1}^N x^i, z)$ for all r, i.e., the game welfare does not depend on which player is allocated to each resource, but only on the total number of players allocated. This is true for the considered SMS rebalancing problem, since the number of daily trips depends only on the total number of vehicles positioned in each region.

In this setting, we define the Equal Share (ES) design choice:

$$f_t^i(x) \stackrel{\text{ES}}{:=} \sum_{r:x[r]>0} \frac{1}{|(x, x_t^{-i})|_r} \cdot \text{ucb}_t^r(x, x_t^{-i}, z_t), \; x \in \mathcal{X}^i. \tag{8}$$

Under ES design, player i's rewards only depend on the resources selected (i.e., where player i has a nonzero allocation) and the welfare from each resource is scaled by the number of players selecting it.

Compared to TW design, the ES rule (8) is computationally more efficient in that each $f_t^i(x)$ is computed using only a subset of the functions $\{\text{ucb}_t^r, r \in [R]\}$. As we will see, it also leads to different performance guarantees. Equal share design was analyzed by Marden & Wierman (2013) in case of a known and separable welfare function. Here, we consider the more challenging scenario where functions γ^r's are unknown. Moreover, we analyze its performance for (generally) non-separable welfare functions considered in this work. To do so, we define the notion of *weak-separability error*.

Given strategy vector $x^i \in \mathcal{X}^i$, we define $[x^i]_{-r}$ to be the modified version of x^i where $x^i[r]$ is set to 0.

Def 4 (Weak-separability error)**.** *We define* weak-separability error *of the game in resource* r *and context* z*,*

$$\epsilon^r(z) = \max_{i \in [N]} \max_{x^i \in \mathcal{X}^i} \gamma^r([x^i]_{-r}, 0, z) - \gamma^r(\mathbf{0}, z). \tag{9}$$

Note that $\epsilon^r(z) \geq 0$, and $\epsilon^r(z) = 0 \; \forall r$ and $\forall z$, when the welfare γ is separable. Moreover, even when γ is non-separable, $\epsilon^r(z) = 0$ in case each player can select at most one resource (i.e., each $x_i \in \mathcal{X}^i$ has at most one nonzero entry). The following theorem (proof in Appendix A.4) bounds the performance of D-SubUCB with ES design.

Thm 2. *Consider the setup of Section 2 and assume the game is anonymous and* $\mathcal{X}^i = \{0, x_{max}\}^R, \forall i$*. When* D-SubUCB *is run with ES design (rule* (8)*) and* β_t^r*'s are set according to Lemma 1 (Appendix A.1), with high probability,*

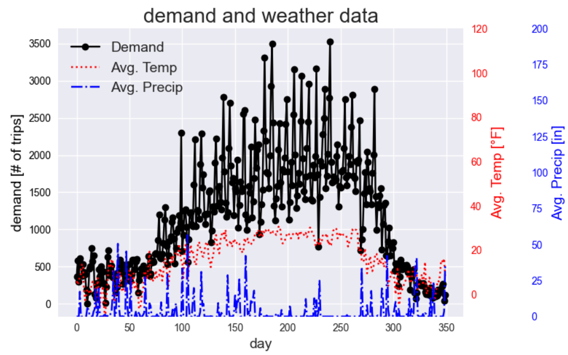
$$\sum_{t=1}^T \gamma(\mathbf{x}_t, z_t) \geq \; \alpha \cdot \text{OPT}$$
$$- \sum_{t=1}^T \sum_{r=1}^R 2\beta_t^r \sigma_t^r(\mathbf{x}_t, z_t) - \sum_{i=1}^N R^i(T) - N \sum_{t=1}^T \sum_{r=1}^R \epsilon^r(z_t),$$

with $\alpha = \max\left\{1 - c_{avg}(\{z_t\}_{t=1}^T), \; \left(1 + c_{wc}(\{z_t\}_{t=1}^T)\right)^{-1}\right\}$*.*

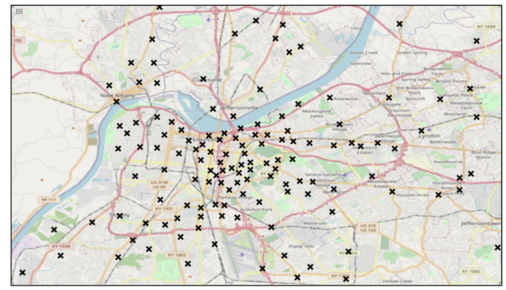
As for TW design (Thm 1), D-SubUCB under ES design approaches a α-approximation of OPT. However, compared to the guarantees of TW design, the standard deviations' term (first term in the second line) in Thm 2 does not depend on the number of players N (under TW design, instead, this term depends linearly on N, see Thm 1). Intuitively, this is because with ES design rule (8) the uncertainty about each function γ^r is shared among the players (while under TW design (6) the reward of each player depends on all such uncertainty). This comes at the price of incurring an extra error term due to the weak-separability errors. Whether this term is sublinear in T (or equal to 0) depends on the considered application. We empirically compare TW and ES design choices in Section 6.

5. Stronger benchmark: Seeking optimal policies

Let us now assume, at each round t, context z_t can be observed *before* choosing allocation $\mathbf{x}_t$. This is the case in many practical scenarios, e.g., when z_t represents time or other seasonal information. In this case, we can consider the stronger performance benchmark of finding the optimal *policy* $\boldsymbol{\pi} : \mathcal{Z} \to \boldsymbol{\mathcal{X}}$ mapping contexts to allocations:



(a) Daily demand and weather data



(b) Candidate drop-off regions

Figure 2. Plot (a) shows the daily demand and weather data (i.e., number of trips, average temperature, and precipitation for the period Jan-December 2019, excluding holidays). Black crosses in (b) indicate the candidate drop-off regions in the city map of Louisville, KY.

$$\mathrm{OPT}_c = \max_{\boldsymbol{\pi}:\mathcal{Z}\to\boldsymbol{\mathcal{X}}} \sum_{t=1}^{T} \gamma(\boldsymbol{\pi}(z_t), z_t)\,. \tag{10}$$

Note that $\mathrm{OPT}_c \geq \mathrm{OPT}$ for all contexts' sequence, as OPT is achieved when one considers only constant policies (i.e., $\boldsymbol{\pi}(z) = \mathbf{x}, \forall z$) in (10).

As we formally show in Appendix B, all the results obtained in the previous sections can also be extended to this richer setting. The difference consists of equipping the N players with algorithms that have sublinear *contextual* regret.

Def 5 (Contextual regret of player i)**.** *The contextual regret of player i after T game rounds is*

$$R_c^i(T) = \max_{\pi:\mathcal{Z}\to\mathcal{X}^i} \sum_{t=1}^{T} f_t^i(\pi(z_t), x_t^{-i}, z_t) - \sum_{t=1}^{T} f_t^i(\mathbf{x}_t, z_t)\,.$$

Several such algorithms exist, depending on assumptions on the rewards, contexts' sequence, and decision sets (see, e.g., Bietti et al., 2018, and references therein). In this scenario, the proposed D-SubUCB simulates a *contextual* game among the N players (as defined by Sessa et al., 2020), and the allocations computed by D-SubUCB satisfy similar performance guarantees to Thm 1 and Thm 2 by replacing OPT with the stronger benchmark OPT_c, and the players' regrets $R^i(T)$ with their contextual counterparts $R_c^i(T)$, under TW and ES design, respectively.

6. Experiments: Learning to rebalance a Shared Mobility System

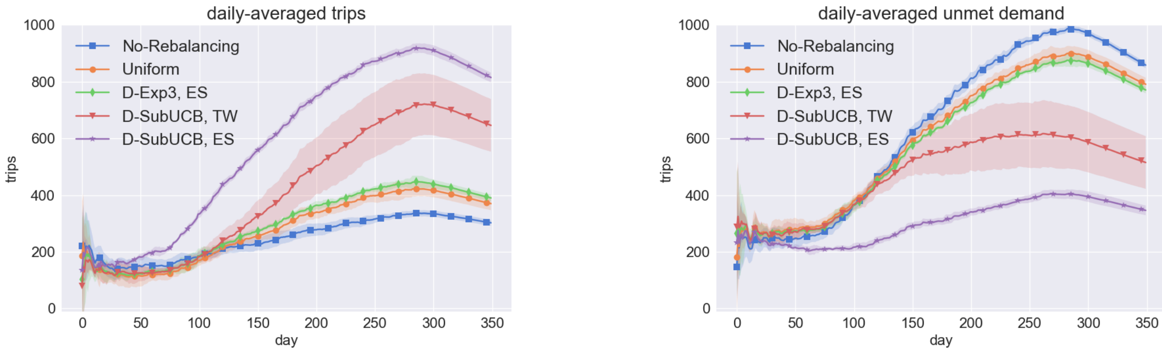
In this section, we evaluate our approach in a realistic case study of rebalancing the SMS of Louisville, KY, based on historical trip data. The system consists of various dockless (i.e., free floating) bike and scooter sharing operators, but we consider it as a unique SMS. We model the rebalancing problem according to the setup of Section 2: before each day t, a rebalancing strategy is represented by $\mathbf{x}_t = [x_t^1, \ldots, x_t^N]$, where $x_t^i[r]$ indicates the number of vehicles truck i drops off in region r. We let the welfare function $\gamma(\mathbf{x}_t, z_t)$ quantify the number of completed trips[4] during day t (each $\gamma^r(\cdot)$ measures the number of trips starting from region r) and evaluate the rebalancing strategies computed by the proposed D-SubUCB algorithm. First, we summarize our data and experimental setup.

Data and experimental setup. Data from Louisville Advanced Planning Office (2020) include trips' timestamps, starting and end coordinates of the dockless SMS of the city of Louisville, KY, for the year of 2019. We use these data to simulate users' demand and trips throughout the one-year period, excluding bank holidays. We also consider weather data (average daily temperature and precipitation) from Weather Underground (2020). We identify $R = 134$ candidate drop-off regions by spatial clustering the trips data using k-means (Lloyd, 1982). We find $k = 300$ initial clusters and iteratively reduce them so that their minimum distance is at least 0.5 km. Figure 2 shows daily demand and weather data (a), and the candidate drop-off regions (b).

Although we have access only to successfully completed trips (met demand), we let the trip data reflect the total users' demand and consider a small number of 40 available vehicles (so that not all the trips can be completed). We consider $N = 5$ trucks, each dropping off 8 vehicles to one of the candidate regions before the day starts (vehicles are positioned at midnight). Hence, $\mathcal{X}^i \subset \{0, 8\}^R, |\mathcal{X}^i| = R$, for $i \in [N]$. We let context $z_t = [z_t[1], z_t[2], z_t[3]] \in \mathbb{R}^3$ represent average daily temperature, precipitation, and the users' demand in day t (i.e., the total number of users willing to rent a vehicle), respectively. Realistically we assume z_t is observed only at the end of each day.

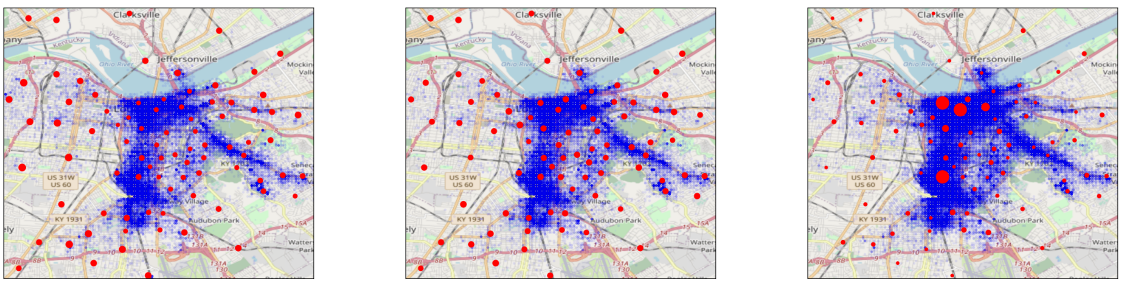
Simulator. Given allocation $\mathbf{x}_t$, the number of daily trips

[4]Our black-box approach allows to model also other performance measures, such as total trips' distance or duration.



(a) Daily-averaged trips (b) Daily-averaged unmet demand

Figure 3. Performance (mean $\pm$ 2 std. out of 3 runs) of different rebalancing strategies. D-SubUCB leads to a higher number of trips and reduces the unmet demand compared to the other baselines.



(a) No-Rebalancing (b) Uniform (c) D-SubUCB with ES design

	(a)	(b)	(c)
# of trips:	106'594	129'281	281'642
unmet trips:	310'739	288'052	135'691

Figure 4. Trips' starting coordinates (blue) and vehicles' allocations (red circles with size proportional to the amount vehicles available in each region at midnight, averaged over the year). When rebalancing does not take place (Left), vehicles tend to concentrate in the outer areas of the city, while under random allocations (Middle) they are uniformly distributed across the city. D-SubUCB learns the users' demand patterns and positions vehicles in strategic areas. This increases the number of successful trips and reduces the unmet demand.

$\gamma(\mathbf{x}_t, z_t)$ is computed as follows. We consider the historical trip data on day t and process them in chronological order. Each of these trips is successful if there exists a region with more than one vehicle whose centroid is within 1 km distance from the trip *start* coordinates. In such a case, one vehicle is moved from this region to the region containing the trip *end* coordinates. Otherwise, the trip is unmet. In Appendix C we formally show that the considered objective is monotone DR-submodular, a fact that we crucially exploit in our proposed approach.

We evaluate the performance of D-SubUCB, under TW and ES design, when each player's no-regret algorithm is MWU (Freund & Schapire, 1997). To learn the welfare functions $\gamma^r(\cdot)$, we use a composite kernel $k(\mathbf{x}_t, z_t) = k_1(\bar{\mathbf{x}}_t, z_t[3]) * k_2(z_t[1], z_t[2])$, where $\bar{\mathbf{x}}_t = \sum_{i=1}^N x_t^i \in \mathbb{R}_+^R$ represents the total number of vehicles positioned in each region, k_1 is a polynomial kernel of degree 3 which measures similarity between allocations and demands, and k_2 is a squared-exponential kernel measuring weather similarity. Moreover, we use two distinct models, depending on day t being a weekday or a weekend. Kernel hyperparameters are optimized offline over 100 random datapoints using a maximum likelihood method and kept fixed for the whole experiment duration. We compare D-SubUCB with the following baselines: 1) No-Rebalancing: Each Sunday night vehicles are randomly distributed over the regions, and until the next Sunday their movement only depends on users' trips (i.e., no rebalancing happens), 2) Uniform: each truck selects a random candidate region at each round t, 3) D-Exp3, a version of D-SubUCB where each player uses the bandit Exp3 (Auer et al., 2003) algorithm (since Exp3 requires only bandit feedback, this baseline does not use RKSH regression to learn the welfare functions and, instead, relies on a high-variance rewards estimator).

In Figure 3 we show that D-SubUCB leads to a higher number of trips and reduces the unmet demand, compared to the baselines. This is also visible from Figure 4, where we plot the average vehicles' allocation (red circles, with size proportional to the average number of vehicles allocated in each region) and the starting coordinates of successful trips (blue circles). Under NO-REBALANCING, vehicles tend to concentrate in the outer areas of the city, while UNIFORM allocates vehicles uniformly over the candidate regions. Instead, after a few days D-SUBUCB learns the users' demand patterns and position vehicles in more strategic zones. This significantly improves upon the bandit baseline D-EXP3 whose high-variance estimator forces a long exploration phase. We also note that ES outperforms TW design. This is in accordance with our theoretical guarantees since according to Section 4 the considered game has separability errors (Def 4) equal to 0 (because each truck can select at most 1 region).

7. Conclusion

We have considered the problem of sequentially allocating agents to a set of resources to maximize a cumulative welfare objective. Different from previous work, we focused on the challenging setting in which the welfare function is unknown, context-dependent, and can only be learned by observing the outcomes of the selected allocations. We have proposed D-SUBUCB, a distributed algorithm that maximizes the cumulative welfare by building and simulating a repeated game among the agents based on upper confidence bounds techniques. Moreover, we have proposed and analyzed two concrete game design choices for our algorithm. Finally, motivated by the recent growth of shared mobility systems, we have demonstrated the effectiveness and practicality of our approach in a realistic case study based on historical trip data of Louisville, KY.

Acknowledgments

This project received funding the Swiss National Science Foundation, under the grant SNSF 200021_172781 and under the NCCR Automation grant 51NF40 180545, by the European Union's ERC grant 815943, and ETH Zürich Postdoctoral Fellowship 19-2 FEL-47.

References

Abbasi-Yadkori, Y. Online learning for linearly parametrized control problems. 2013.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2003.

Bach, F. Submodular functions: From discrete to continuous domains. *Mathematical Programming*, 175(1–2): 419–459, 2019.

Bhatia, A., Varakantham, P., and Kumar, A. Resource constrained deep reinforcement learning. *International Conference on Automated Planning and Scheduling*, pp. 610–620, 2019.

Bian, A., Levy, K., Krause, A., and Buhmann, J. M. Continuous DR-submodular maximization: Structure and algorithms. In *Neural Information Processing Systems (NeurIPS)*, volume 30, pp. 486–496, 2017.

Bietti, A., Agarwal, A., and Langford, J. A contextual bandit bake-off. *ArXiv*, abs/1802.04064, 2018.

Blum, A., Hajiaghayi, M., Ligett, K., and Roth, A. Regret minimization and the price of total anarchy. In *Annual ACM Symposium on Theory of Computing*, STOC '08, pp. 373–382, 2008.

Bogunovic, I., Scarlett, J., Jegelka, S., and Cevher, V. Adversarially robust optimization with Gaussian processes. In *Neural Information Processing Systems (NeurIPS)*, volume 31, pp. 5760–5770, 2018.

Cesa-Bianchi, N. and Lugosi, G. *Prediction, learning, and games*. Cambridge University Press, 2006.

Chen, L., Krause, A., and Karbasi, A. Interactive submodular bandit. In *Neural Information Processing Systems (NeurIPS)*, volume 30, pp. 141–152, 2017.

Chowdhury, S. R. and Gopalan, A. On kernelized multi-armed bandits. In *International Conference on Machine Learning (ICML)*, 2017.

Dell'Amico, M., Hadjicostantinou, E., Iori, M., and Novellani, S. The bike sharing rebalancing problem: Mathematical formulations and benchmark instances. *Omega*, 45:7 – 19, 2014.

Frazier, P. I. A tutorial on Bayesian optimization. *ArXiv*, abs/1807.02811, 2018.

Freund, D., Norouzi-Fard, A., Paul, A., Wang, C., Henderson, S. G., and Shmoys, D. B. *Data-Driven Rebalancing Methods for Bike-Share Systems*, pp. 255–278. 2020.

Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

Ghosh, S., Varakantham, P., Adulyasak, Y., and Jaillet, P. Dynamic redeployment to counter congestion or starvation in vehicle sharing systems. pp. 79–87, 2015.

Golovin, D., Krause, A., and Streeter, M. Online submodular maximization under a matroid constraint with application to learning assignments. *ArXiv*, abs/1407.1082, 2014.

Jian, N., Freund, D., Wiberg, H. M., and Henderson, S. G. Simulation optimization for a large-scale bike-sharing system. In *Winter Simulation Conference*, pp. 602–613, 2016.

Katoh, N. and Ibaraki, T. *Resource Allocation Problems*, pp. 905–1006. Springer US, Boston, MA, 1998.

Krause, A. and Guestrin, C. Near-optimal observation selection using submodular functions. In *National Conference on Artificial Intelligence - Volume 2*, AAAI'07, pp. 1650–1654, 2007.

Krause, A. and Ong, C. Contextual Gaussian process bandit optimization. In *Neural Information Processing Systems (NeurIPS)*, volume 24, pp. 2447–2455, 2011.

Laporte, G., Meunier, F., and Calvo, R. W. Shared mobility systems: an updated survey. *Annals of Operations Research*, 271(1):105–126, December 2018.

Lin, L., He, Z., and Peeta, S. Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach. *Transportation Research Part C: Emerging Technologies*, 97:258 – 276, 2018.

Lloyd, S. P. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28:129–137, 1982.

Louisville Advanced Planning Office. Dockless vehicles data, 2020. `https://data.louisvilleky.gov/dataset/dockless-vehicles`.

Marden, J. R. and Wierman, A. Distributed welfare games. *Operations Research*, 61(1):155–168, 2013.

Mockus, J. *Bayesian Approach to Global Optimization: Theory and Applications*. Springer Netherlands, Dordrecht, 1989.

Paccagnan, D., Chandan, R., and Marden, J. R. Utility design for distributed resource allocation—part i: Characterizing and optimizing the exact price of anarchy. *IEEE Transactions on Automatic Control*, 65(11):4616–4631, 2020.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

Sessa, P. G., Bogunovic, I., Kamgarpour, M., and Krause, A. No-regret learning in unknown games with correlated payoffs. In *Proc. Neural Information Processing Systems (NeurIPS)*, December 2019a.

Sessa, P. G., Kamgarpour, M., and Krause, A. Bounding inefficiency of equilibria in continuous actions games using submodularity and curvature. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019b.

Sessa, P. G., Bogunovic, I., Krause, A., and Kamgarpour, M. Contextual games: Multi-agent learning with side information. In *Proc. Neural Information Processing Systems (NeurIPS)*, December 2020.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*, 2010.

Stanczak, S., Wiczanowski, M., and Boche, H. *Resource Allocation in Wireless Networks: Theory and Algorithms (Lecture Notes in Computer Science)*. Springer-Verlag, Berlin, Heidelberg, 2006.

Vetta, A. Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In *Symposium on Foundations of Computer Science*, FOCS '02, pp. 416–, 2002.

Weather Underground. Weather data, 2020. `https://www.wunderground.com`.

Yang, Z., Hu, J., Shu, Y., Cheng, P., Chen, J., and Moscibroda, T. Mobility modeling and prediction in bike-sharing systems. *International Conference on Mobile Systems, Applications, and Services*, 2016.

Zhang, M., Chen, L., Hassani, H., and Karbasi, A. Online continuous submodular maximization: From full-information to bandit feedback. In *Neural Information Processing Systems (NeurIPS)*, volume 32, pp. 9210–9221, 2019.