Collaborative Bayesian Optimization with Fair Regret

Rachael Hwee Ling Sim [*1] **Yehong Zhang** [*2] **Bryan Kian Hsiang Low** [1] **Patrick Jaillet** [3]

Abstract

Bayesian optimization (BO) is a popular tool for optimizing complex and costly-to-evaluate black-box objective functions. To further reduce the number of function evaluations, any party performing BO may be interested to collaborate with others to optimize the same objective function concurrently. To do this, existing BO algorithms have considered optimizing a batch of input queries in parallel and provided theoretical bounds on their cumulative regret reflecting inefficiency. However, when the objective function values are correlated with real-world rewards (e.g., money), parties may be hesitant to collaborate if they risk incurring larger cumulative regret (i.e., smaller real-world reward) than others. This paper shows that *fairness* and *efficiency* are both necessary for the collaborative BO setting. Inspired by social welfare concepts from economics, we propose a new notion of regret capturing these properties and a collaborative BO algorithm whose convergence rate can be theoretically guaranteed by bounding the new regret, both of which share an adjustable parameter for trading off between fairness vs. efficiency. We empirically demonstrate the benefits (e.g., increased fairness) of our algorithm using synthetic and real-world datasets.

1. Introduction

Bayesian optimization (BO) (Dai et al., 2019; 2020a;b; Ling et al., 2016; Zhang et al., 2019) is a popular tool for optimizing complex (e.g., possibly noisy, non-convex, and/or with no closed-form expression/derivative) and costly-to-evaluate black-box objective functions given a limited budget. BO algorithms can achieve competitive optimization performance by sequentially selecting input queries for evaluating the objective function f to trade off between sampling at or close to a likely maximizer (i.e., exploitation) vs. sampling in an unobserved input region (i.e., exploration). BO has been used in a wide variety of real-world applications such as hyperparameter tuning of *machine learning* (ML) models (Shahriari et al., 2015), drug development, chemical/material design (Griffiths & Hernández-Lobato, 2020; Zhang et al., 2020), hotspot discovery via a *mobility-on-demand* (MoD) system with *autonomous vehicles* (AVs) (Chen et al., 2013; 2015; Kharkovskii et al., 2020), among others. In these applications, each function evaluation (e.g., evaluating molecular properties, MoD AV cruising to a location) is economically costly and/or time-consuming. Naturally, any party who performs BO would be interested in reducing the number of function evaluations and hence the cost/time needed. One appealing solution is collaborative ML (Sim et al., 2020) which allows every party to share data from function evaluations with the other parties optimizing the *same* f concurrently instead of evaluating the expensive f on its own. The works of Cheng et al. (2020) and Fitch (2018) have discussed the importance and benefits of collaboration in drug discovery and precision agriculture.

At first glance, existing *batch* BO algorithms (Daxberger & Low, 2017; Desautels et al., 2014; Wu & Frazier, 2016) may seem like a suitable solution. In each iteration, these algorithms select a batch of input queries and each input query is assigned to a collaborating party for evaluating f in *parallel*. However, these batch BO algorithms do not optimize the assignment of input queries to parties. Some parties may be unintentionally assigned input queries to evaluate f with only small values. This possibility discourages collaboration and data sharing. *Why is this so?* Each party usually incurs some cost to evaluate f and is particular about its value as it is often correlated with some real-world reward. Any party would not want to *unfairly* receive smaller real-world reward and less valuable information than other collaborating parties. For example, when companies operating MoD AVs collaborate to discover mobility demand hotspots, a party evaluating at a location with a higher mobility demand is more likely to pick up passengers and earn fares. So, any MoD company would not want to solely evaluate at locations with low mobility demand and altruistically provide information to the other parties. When farmers collaborate

[*]Equal contribution [1]Department of Computer Science, National University of Singapore, Republic of Singapore [2]Peng Cheng Laboratory, People's Republic of China [3]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, USA. Correspondence to: Bryan Kian Hsiang Low <lowkh@comp.nus.edu.sg>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

to optimize the conditions (e.g., fertilizer composition) to grow crops, a farmer would not like to always test conditions with poor yield to benefit others. Likewise, a research group collaborating with others in chemical or drug design would not want to study significantly fewer or less promising molecules than the others.

Then, *what guarantees should a collaborative BO algorithm provide to address every party's concerns?* Firstly, the algorithm must have good *efficiency*: It should be able to maximize the evaluated function values across all parties (i.e., cumulative utility) and simultaneously find the maximizer. The *cumulative utility* (CU) here is closely related to the notion of *cumulative regret* used to measure the optimization performance of several BO algorithms (Daxberger & Low, 2017; Desautels et al., 2014; Hoang et al., 2018; Srinivas et al., 2010). Next, to avoid the unfair real-world rewards in the aforementioned examples, the algorithm needs to ensure *fairness* by reducing differences between the CUs of all parties. However, incorporating both *efficiency* and *fairness* in a BO algorithm is nontrivial due to the lack of a formal notion of fairness and the trade-off between them. The trade-off arises as fairness can be achieved by preventing some parties from getting larger CUs. For example, the fair strategy of selecting the same input queries for all parties may hurt efficiency due to redundant function evaluations.

To address this challenge, our work here considers a collaborative BO mechanism where a trusted mediator jointly selects the input queries for all parties based on their data, as in batch BO algorithms. A new notion of regret is defined based on a social welfare concept from economics for capturing both inefficiency and unfairness. We then design a collaborative BO algorithm whose convergence rate can be theoretically guaranteed by bounding the new regret, i.e., the algorithm would not produce inefficient and unfair assignments. This is novelly achieved by considering each party's CU up to the previous BO iteration when selecting input queries in the current BO iteration to reduce unfairness. The specific contributions of our work here include:

- Defining new notions of *fair* regret based on a social welfare concept economics called the *generalized Gini social-evaluation function* (G2SF) (Weymark, 1981) that considers both efficiency and fairness (Sec. 3);
- Proposing a collaborative BO algorithm that can theoretically guarantee its convergence rate by bounding the new fair regret, achieve asymptotic *no-regret* performance, and address the concern of collaborative fairness (Sec. 4);
- Designing an adjustable parameter that can be used to trade off between fairness vs. efficiency in both the new fair regret and our collaborative BO algorithm (Sec. 3);
- Demonstrating the increased fairness and other properties of our collaborative BO algorithm empirically (Sec. 5).

To the best of our knowledge, this is the first algorithm that addresses collaborative fairness in BO and considers fairness in the cumulative sense. The only other *fair BO* work (Perrone et al., 2020) has focused on mitigating biases in the outputs, which is a different line of fairness concept. More literature related to fairness (including the various fairness concepts in multi-armed bandit) and incentivizing exploration will be discussed later in Sec. 6.

2. Problem Formulation and Background

Our problem setting considers n parties jointly optimizing a black-box objective function $f : \mathcal{X} \rightarrow \mathbb{R}$ where $\mathcal{X} \subset \mathbb{R}^d$ denotes a domain of d-dimensional input feature vectors.[1] We assume that each party can evaluate the objective function f at any input $\mathbf{x} \in \mathcal{X}$ and the availability of a *trusted mediator*[2] whom every *honest* party is willing to share its data with. In each iteration $t = 1, \ldots, T$, the mediator selects an input query $\mathbf{x}_t^i$ to be assigned to each party i who then evaluates f at the assigned input $\mathbf{x}_t^i$ to observe a noisy realized output $y_t^i \triangleq f(\mathbf{x}_t^i) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ with noise variance σ^2. Let $[n] \triangleq \{1, \ldots, n\}$. Let $\mathbf{X}_t \triangleq (\mathbf{x}_t^i)_{i \in [n]}$ and $\mathbf{y}_t \triangleq (y_t^i)_{i \in [n]}^\top$ be an input matrix and the corresponding vector of noisy realized outputs in iteration t, respectively. After t iterations, the mediator has a pooled dataset $D_{1:t} \triangleq (\mathbf{X}_{1:t}, \mathbf{y}_{1:t})$ where $\mathbf{X}_{1:t} \triangleq (\mathbf{X}_j)_{j=1,\ldots,t}$ and $\mathbf{y}_{1:t} \triangleq (\mathbf{y}_j)_{j=1,\ldots,t}$. The number T of iterations can be decided in advance or determined when any party wishes to leave the collaboration.

2.1. Batch Bayesian Optimization (BO)

The objective of a conventional batch BO is to find a global maximizer $\mathbf{x}^* \triangleq \arg\max_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$ of the objective function f. To achieve this, the black-box objective function f is modeled as a *Gaussian process* (GP), that is, every finite subset of $\{f(\mathbf{x})\}_{\mathbf{x} \in \mathcal{X}}$ follows a multivariate Gaussian distribution (Rasmussen & Williams, 2006). A GP is fully specified by its prior mean $m_{\mathbf{x}} \triangleq \mathbb{E}[f(\mathbf{x})]$ and covariance $k_{\mathbf{x}\mathbf{x}'} \triangleq \mathrm{cov}[f(\mathbf{x}), f(\mathbf{x}')]$ for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. In iteration t, a GP model can produce a predictive distribution of the function outputs $\mathbf{f}_{\mathbf{X}_t} \triangleq (f(\mathbf{x}_t^i))_{i \in [n]}$ at any input matrix $\mathbf{X}_t$: $p(\mathbf{f}_{\mathbf{X}_t} | D_{1:t-1}) = \mathcal{N}(\mu_{\mathbf{X}_t | D_{1:t-1}}, \Sigma_{\mathbf{X}_t | D_{1:t-1}})$ where $\mu_{\mathbf{X}_t | D_{1:t-1}}$ and $\Sigma_{\mathbf{X}_t | D_{1:t-1}}$ are, respectively, the predictive/posterior mean vector and covariance matrix which can be computed analytically (Hoang et al., 2015; 2016).

Then, in each iteration t, a batch BO algorithm selects a batch of inputs $\mathbf{X}_t \in \mathcal{X}^n$ to maximize some acquisition function that is computed using the predictive distribution

[1] Our theoretical analysis considers a discrete input domain. However, our results can be generalized to a continuous, compact input domain via a suitable discretization (Srinivas et al., 2010).

[2] In reality, the mediator can be a government agency or an industry association who acts in everyone's best interest equally.

of $\mathbf{f}_{\mathbf{X}_t}$. For example, the *distributed batch GP upper confidence bound* (DB-GP-UCB) algorithm (Daxberger & Low, 2017) selects $\mathbf{X}_t$ that trades off between sampling inputs with large posterior mean (i.e., exploitation) vs. those with high information gain on $\mathbf{f}_{\mathcal{X}}$ (i.e., exploration):

$$\max_{\mathbf{X}_t \in \mathcal{X}^n} \mathbf{1}^\top \mu_{\mathbf{X}_t | D_{1:t-1}} + \sqrt{\alpha_t \mathbb{I}[\mathbf{f}_{\mathcal{X}}; \mathbf{y}_t \mid D_{1:t-1}]} \quad (1)$$

where parameter α_t is set to trade off between exploitation vs. exploration for bounding its cumulative regret (Sec. 2.2), while $\mathbb{I}[\mathbf{f}_{\mathcal{X}}; \mathbf{y}_t \mid D_{1:t-1}] = 0.5 \log \mid I + \sigma^{-2} \Sigma_{\mathbf{X}_t | D_{1:t-1}} \mid$ is the information gain (or, equivalently, reduction in uncertainty) on $\mathbf{f}_{\mathcal{X}}$ by evaluating f at $\mathbf{X}_t$ to observe $\mathbf{y}_t$ given the dataset $D_{1:t-1}$ from previous iterations $1, \ldots, t-1$, and is larger when the inputs within $\mathbf{X}_t$ are more diverse and dissimilar to the inputs $\mathbf{X}_{1:t-1}$ from earlier iterations.

2.2. Regret and Utility

The above DB-GP-UCB algorithm is designed to minimize the *full cumulative regret* which is a common BO objective. Let $r_t^i \triangleq f(\mathbf{x}^*) - f(\mathbf{x}_t^i)$ be the *instantaneous regret* incurred by assigning $\mathbf{x}_t^i$ to party i in iteration t. The *full instantaneous regret* is the sum of instantaneous regrets incurred by all parties of the batch in iteration t: $r_t \triangleq \sum_{i=1}^n r_t^i$. The full cumulative regret is the sum of full instantaneous regrets over iterations $t = 1, \ldots, T$: $R_T \triangleq \sum_{t=1}^T r_t$. When a BO algorithm asymptotically achieves *no regret* (i.e., $\lim_{T\to\infty} R_T/T = 0$), it will eventually converge to a global maximum and every party will evaluate f at the assigned maximizer $\mathbf{x}^*$.

Let the *utility* $u_t^i \triangleq f(\mathbf{x}_t^i)$ achieved by party i in iteration t be the function output $f(\mathbf{x}_t^i)$ that it has evaluated at the assigned $\mathbf{x}_t^i$. The *full utility* is the sum of utilities achieved by all parties of the batch in iteration t: $u_t \triangleq \sum_{i=1}^n u_t^i$. The *full cumulative utility* is the sum of full utilities over iterations $t = 1, \ldots, T$: $U_T \triangleq \sum_{t=1}^T u_t$. Both u_t and U_T are considered measures of *efficiency*. The full cumulative regret equals to the loss in utility from not knowing $\mathbf{x}^*$ beforehand: $R_T = nTf(\mathbf{x}^*) - U_T$. So, minimizing R_T is equivalent to maximizing U_T. Though R_T is more commonly used than U_T in the BO literature, we have introduced the notion of utility to ease the discussion of the concept of fairness.

The properties formalized below are satisfied by r_t and u_t. Such properties should also be satisfied by alternative notions of regret and utility so that achieving no regret would imply convergence to a global maximum for every party:

E1 **Monotonicity.** If the utility of any party improves in any iteration t, *ceteris paribus*, then u_t should increase and r_t should decrease: Let $\{u_t^i\}_{i\in[n]}$ and $\{\hat{u}_t^i\}_{i\in[n]}$ denote any two sets of utilities achieved by parties $i \in [n]$ in iteration t, and r_t (u_t) and $\hat{r}_t$ ($\hat{u}_t$) be the corresponding full instantaneous regrets (full utilities). For $t = 1, \ldots, T$,

$$\forall i \in [n] \;\; (u_t^i > \hat{u}_t^i) \wedge (\forall j \in [n] \setminus \{i\} \; u_t^j = \hat{u}_t^j) \Rightarrow (u_t > \hat{u}_t) \;\wedge\; (r_t < \hat{r}_t) \;.$$

E2 **Instantaneity.** The full instantaneous regret in iteration t is 0 if every party $i \in [n]$ is assigned $\mathbf{x}^*$ to evaluate f: For $t = 1, \ldots, T$,

$$(\forall i \in [n] \;\; \mathbf{x}_t^i = \mathbf{x}^*) \Rightarrow r_t = 0 \;.$$

2.3. Social Welfare and Fairness

In this subsection, we will introduce a number of social welfare and fairness concepts from economics which will be referenced later in this paper. Suppose that there are n parties and each party $i \in [n]$ has a utility a^i. Let $\boldsymbol{a} \triangleq (a^i)_{i\in[n]}$ be a utility vector. A *social welfare* function projects a utility vector to a real value and naturally induces an *ordering* ($\preceq$) of the utility vectors: A utility vector $\boldsymbol{b}$ is preferred over $\boldsymbol{a}$ (i.e., denoted as $\boldsymbol{a} \preceq \boldsymbol{b}$) if the social welfare $SW(\boldsymbol{b})$ of $\boldsymbol{b}$ is larger than the social welfare $SW(\boldsymbol{a})$ of $\boldsymbol{a}$: $SW(\boldsymbol{a}) < SW(\boldsymbol{b})$. Some commonly considered social welfare functions and orderings (Endriss, 2018) are listed below:

Utilitarian social welfare (USW) is defined as a sum of the utilities of all parties $i \in [n]$: $SW_{\text{util}}(\boldsymbol{a}) \triangleq \sum_{i=1}^n a^i$. Maximizing the USW maximizes the averaged utility over all parties, but it does not consider fairness. For example, given $n = 3$, the utility vector $(3, 3, 3)$ is obviously fairer than $(1, 1, 7)$, but their USWs are the same. The full cumulative utility U_T defined in Sec. 2.2 is a form of USW.

Egalitarian (max-min) social welfare (ESW) is defined as the minimum utility of any party $i \in [n]$: $SW_{\text{egal}}(\boldsymbol{a}) \triangleq \min_{i\in[n]} a^i$. ESW prefers the fairer outcome of $(3, 3, 3)$ over $(1, 1, 7)$.

Leximin ordering is an ordering method that cannot be induced from any social welfare function. A utility vector $\boldsymbol{b}$ is preferred over $\boldsymbol{a}$ in a Leximin ordering (i.e., denoted as $\boldsymbol{a} \preceq_{\text{lex}} \boldsymbol{b}$) iff $\boldsymbol{a}$ and $\boldsymbol{b}$ are identical, or after sorting in ascending order to obtain $\boldsymbol{a}_*$ and $\boldsymbol{b}_*$, there exists an $i \in [n]$ s.t. $a_*^i < b_*^i$ and $a_*^j = b_*^j$ for all $j < i$. Note that this implies $SW_{\text{egal}}(\boldsymbol{a}) \leq SW_{\text{egal}}(\boldsymbol{b})$. Leximin ordering extends the idea of maximizing the utility of the worst-off party in ESW to the second worst-off party next, then the third, and so on.

Next, how do we decide if the ordering $\boldsymbol{a} \preceq \boldsymbol{b}$ is fair? Intuitively, given a utility vector $\boldsymbol{a}$, if a party with a higher utility transfers $\leq 1/2$ of its excess utility to another worse-off party, then it will result in a new utility vector $\boldsymbol{b}$ with more similar utilities and so, $\boldsymbol{b}$ should be preferred for fairness. This central intuition on fairness is captured by the ***Pigou-Dalton principle*** **(PDP)** (Dalton, 1920; Pigou, 1912). Formally, a social welfare function or an ordering satisfies PDP in a *strong* sense if $\boldsymbol{b}$ is preferred over $\boldsymbol{a}$ (i.e., $\boldsymbol{a} \prec \boldsymbol{b}$) whenever there exist $i, j \in [n]$ s.t. (a) $\forall k \in [n] \setminus \{i, j\} \; a^k = b^k$,

(b) $a^i + a^j = b^i + b^j$, and (c) $|a^i - a^j| > |b^i - b^j|$. To satisfy the PDP in the *weak* sense, we only require that $\boldsymbol{a} \preceq \boldsymbol{b}$. For an *unfairness* metric represented by a function H to satisfy the PDP, we will instead require $H(\boldsymbol{a}) > H(\boldsymbol{b})$ which means that the unfairness in $\boldsymbol{a}$ is more than that in $\boldsymbol{b}$. This general concept of fairness gives us flexibility in selecting a fair and efficient notion of social welfare.

3. Notions of Fair Regret in Collaborative BO

Recall that we consider n parties jointly optimizing a common black-box objective function f. A mediator repeatedly selects separate input queries to be assigned to different parties who then evaluate f at these assigned inputs. During the BO process, each party shares its data with the mediator only and does not know the data of the other parties. Such an information asymmetry process provides two key implications. Firstly, each party prefers to stay in the collaboration instead of leaving to optimize f independently because evaluating f at the inputs assigned by the mediator is the only way to benefit from more data gathered by all the parties.[3] Next, it prevents any self-interested party from evaluating f at inputs with large function values/real-world rewards discovered by the other parties, which may increase unfairness. Moreover, it protects the privacy of the parties (e.g., location history of MoD AV). To incentivize any self-interested party (e.g., an MoD company or a research group) to join the collaboration, our work proposes a collaborative BO algorithm that considers both efficiency and fairness when selecting the input queries. To do this, the conventional batch BO objective of minimizing the full cumulative regret R_T (Sec. 2.2) is no longer suitable since it is equivalent to maximizing the full cumulative utility which is a form of USW (Sec. 2.3) — no fairness is considered during the BO process. A new notion of regret, which incorporates both efficiency and fairness, is needed.

Following the notations in Sec. 2.2, let $U_T^i \triangleq \sum_{t=1}^{T} u_t^i$ be the *individual cumulative utility* (CU) of party i, and $\boldsymbol{u}_T \triangleq (U_T^i)_{i\in[n]}$ be a vector of individual CUs of all parties. We can define a *generalized* CU by aggregating the individual CUs of all parties in a utility vector $\boldsymbol{u}_T$ using a function W which projects $\boldsymbol{u}_T$ to a real value: $GU_T \triangleq W(\boldsymbol{u}_T)$. Note that $GU_T = U_T$ when W is set as USW. In our work here, we propose a suitable function W such that maximizing W corresponds to (a) maximizing the individual CU of each party (i.e., efficiency) and (b) reducing the differences between the individual CUs of all parties (i.e., fairness). To achieve this, W can be any welfare function that satisfies both monotonicity (E1) and the PDP. We will also define W using the *generalized Gini social-evaluation function* (G2SF) since it is more general than the welfare functions described in Sec. 2.3 and, more importantly, provides an adjustable parameter that can be used to trade off between efficiency vs. fairness, as detailed in Sec 3.2.

[3]We will show this implication later via empirical evaluation. A rigorous mechanism that can incentivize the parties to adhere to the mediator and stay in the collaboration is left to future work.

3.1. Fair Regret

Given the generalized CU (i.e., GU_T), we can define a *generalized* cumulative regret similar to that described in Sec. 2.2: $GR_T \triangleq W(T\boldsymbol{f}^*) - GU_T$ where $\boldsymbol{f}^* \triangleq f(\boldsymbol{x}^*)\mathbf{1}$. Though it seems natural and ideal to consider minimizing GR_T directly, it is challenging to do so within the iterative BO process since $GU_T = W(\boldsymbol{u}_T)$ depends on the dataset of every party in all the BO iterations and may not be decomposable into independent terms across iterations $t = 1, \ldots, T$ for certain choices of W, unlike U_T that can be decomposed into additive terms $u_1, \ldots, u_T$.

To tackle this challenge, we propose to *greedily* optimize the trade-off between efficiency vs. fairness in each BO iteration instead. Specifically, let $U_t^i \triangleq \sum_{t'=1}^{t} u_{t'}^i$ be the *individual t-step cumulative utility* (t-CU) of party i for $t = 1, \ldots, T$ and $\boldsymbol{u}_t \triangleq (U_t^i)_{i\in[n]}$ be a vector of individual t-CUs of all parties. Correspondingly, let the generalized t-CU be $g_t \triangleq W(\boldsymbol{u}_t)$. We define a new notion of instantaneous *fair regret* s_t by subtracting g_t from the maximally achievable value in iteration t:

$$
\begin{aligned}
s_t &\triangleq W(\boldsymbol{f}^* + \boldsymbol{u}_{t-1}) - g_t \\
&= W((f(\mathbf{x}^*) + U_{t-1}^i)_{i\in[n]}) - W((f(\mathbf{x}_t^i) + U_{t-1}^i)_{i\in[n]})
\end{aligned} \tag{2}
$$

where the equality is due to the definition of $\boldsymbol{u}_t$ and $u_t^i = f(\mathbf{x}_t^i)$. To understand (2), for $t = 1, \ldots, T$, the maximally achievable value of the individual t-CU of any party $i \in [n]$ is $f(\mathbf{x}^*) + U_{t-1}^i$ instead of $tf(\mathbf{x}^*)$ as its individual $(t-1)$-CU U_{t-1}^i achieved in previous iterations $1, \ldots, t-1$ cannot be changed. This new fair regret satisfies instantaneity (E2 in Sec. 2.2) since $s_t = 0$ if $\mathbf{x}_t^i = \mathbf{x}^*$ for all $i \in [n]$. Then, a *fair cumulative regret* can be defined as $S_T \triangleq \sum_{t=1}^{T} s_t$.

From Sec. 2, before iteration t, any party i can only observe the noisy realized outputs $(y_{t'}^i)_{t'=1,\ldots,t-1}^\top$ instead of the noiseless function outputs $(f(\mathbf{x}_{t'}^i))_{t'=1,\ldots,t-1}^\top$. Consequently, since each party i does not know the value of U_{t-1}^i, U_{t-1}^i should not be directly used in our algorithm. Instead, it is reasonable to use the noisy *realized/observed* variant of the individual $(t-1)$-CU: $\lambda_t^i \triangleq \sum_{t'=1}^{t-1} y_{t'}^i$ as the parties are particular about observing the realized outputs (e.g., mobility demand correlating with the resulting income) in the real-world applications.[4] So, we can modify s_t to obtain a revised notion of instantaneous fair regret s'_t instead:

$$
s'_t \triangleq W((f(\mathbf{x}^*) + \lambda_t^i)_{i\in[n]}) - W((f(\mathbf{x}_t^i) + \lambda_t^i)_{i\in[n]}) \tag{3}
$$

[4]In practice, when f is unknown/black-box, R_T and U_T are computed by also approximating $f(\mathbf{x}_t^i)$ with y_t^i.

which still satisfies instantaneity (E2) since $s'_t = 0$ if $\mathbf{x}^i_t = \mathbf{x}^*$ for all $i \in [n]$. Similarly, let $g'_t \triangleq W((f(\mathbf{x}^i_t) + \lambda^i_t)_{i\in[n]})$.

Remark 1. We consider fairness in individual t-CUs among the parties in *every* BO iteration $t = 1, \ldots, T$ instead of only at the final iteration T. This is desirable for real-world applications as the parties may be concerned about fairness *during* the BO process and may not fix T in advance.

Remark 2. *Cooperative game theory* (CGT) (Chalkiadakis et al., 2011) is another framework that may be incorporated into BO to achieve fairness by distributing/assigning the rewards/input queries to all parties according to their data valuations. Our work here chooses to adopt social welfare concepts instead of CGT as the social welfare functions are cheaper to compute and easier to incorporate into BO than the data valuation functions (e.g., Shapley value) from CGT. However, we will empirically study a CGT concept known as *individual rationality* which states that each party should not be worse off via collaborating than working alone.

3.2. Generalized Gini Social-Evaluation Function

Our above notion of fair regret (3) is defined over a general welfare function W. We will next introduce the specific form of W that is used in our BO algorithm. The Gini index is a popular measure of income inequality and is related to the *Gini social-evaluation function* (GSF). A fairer distribution of incomes should have a smaller Gini index and larger GSF value. Formally, let $\langle w_1, \ldots, w_n\rangle$ be a sequence of positive and non-increasing weights, $\mathbf{w} \triangleq (w_i)_{i\in[n]}$ be a weight vector, and ϕ be a sorting operator that sorts the elements of its input vector (e.g., $\boldsymbol{u}_t$) in an ascending order and returns a new sorted vector. GSF is a subclass of *ordered weighted average* functions. According to GSF, we can define the generalized t-CU as

$$g_t \triangleq W(\boldsymbol{u}_t) = \mathbf{w}^\top \phi(\boldsymbol{u}_t) \tag{4}$$

s.t. its corresponding Gini weights are set as odd numbers in a descending order: $w_i = 2(n-i) + 1$ for all $i \in [n]$. Intuitively, (4) assigns a larger weight w_i to a smaller individual t-CU U^i_t and outputs a larger g_t for a utility vector $\boldsymbol{u}_t$ whose elements are more similar, which aligns with our intuition that such values of individual t-CUs of all parties are fairer and more desirable. For example, given 2 parties, the g_t with utility vector $(5, 5)$ is $W(5, 5) = 20$ which is larger than g_t with $(8, 2)$ or $(2, 8)$ (i.e., $3\times 2+1\times 8 = 14$). However, both their CUs are equal (i.e., $5 + 5 = 8 + 2$).

As there is no strong reason for the Gini weights or any other choice of weights, the work of Weymark (1981) has generalized W to the *generalized Gini social-evaluation function* (G2SF) which allows an arbitrary sequence of positive non-increasing weights. Positive weights address "monotonicity" (E1), while pairing non-increasing (decreasing) weights with increasing utilities has been shown to satisfy the PDP in the weak (strong) sense (see Axiom 6 in (Weymark, 1981) for more details) and thus addresses "fairness". The weights $\mathbf{w}$ can be used to trade off between efficiency vs. fairness, as discussed later in Sec. 3.3.

3.3. Efficiency vs. Fairness in the Fair Regret

When W is set as a G2SF function, the weights $\mathbf{w}$ can be used to trade off between efficiency vs. fairness in g'_t (or s'_t) and g_t (or s_t). For simplicity, we will focus our discussion in this section on g_t (or s_t) but it would also apply to g'_t (or s'_t). To devise a way to control the efficiency vs. fairness trade-off, we start by examining two extreme cases:

- If $w_i = 1$ for all $i \in [n]$, then g_t reduces to USW (Sec. 2.3). Maximizing g_t would lead to maximum efficiency (i.e., largest U_T). The fair cumulative regret is $S_T = R_T$ since in each of its additive terms s_t (2), U^i_{t-1} in its first and second terms would cancel out.
- If $w_1 = 1$ and $w_i = 0$ for $i = 2, \ldots, n$, then g_t reduces to ESW (Sec. 2.3) which only considers the utility of the worst-off party. In this case, s_t does not satisfy monotonicity (E1 in Sec. 2.2) as the utilities of the other $n-1$ parties are assigned a weight of 0 and hence ignored.

In the first case, the weights are identical, while in the second case, the relative difference between w_1 vs. w_i for $i = 2, \ldots, n$ is large. Intuitively, smaller relative differences between w_i's would cause the larger weighted utilities to make up a bigger proportion of g_t and hence cause g_t to induce an ordering that may prefer more efficient but less fair utility vectors. To better control the differences between the non-increasing w_i for all $i \in [n]$, we set $w_i = \rho^{i-1}$ with $0 < \rho \leq 1$ s.t. a single parameter ρ can be used to trade off between efficiency vs. fairness, as detailed below:

Proposition 1. *Set $w_i = \rho^{i-1}$ for all $i \in [n]$ with $0 < \rho \leq 1$. Then, s_t will satisfy monotonicity* (E1) *and the* Pigou-Dalton principle *(PDP) on fairness. Moreover, when*

- *$\rho = 1$, W is USW s.t. g_t and s_t only satisfy the PDP in the weak sense;*
- *$0 < \rho < 1$, g_t and s_t satisfy PDP in the strong sense;*
- *$\rho \to 0$, $w_i/w_1 \to 0$ for $i = 2, \ldots, n$, and the preference of utility vectors induced by g_t converges to that of ESW and Leximin ordering: $\boldsymbol{a} \preceq_{lex} \boldsymbol{b} \Leftrightarrow W(\boldsymbol{a}) < W(\boldsymbol{b})$.*

Its proof is in Appendix B.1. Informally, Proposition 1 holds as $\langle w_1, \ldots, w_n\rangle$ forms a decreasing sequence that satisfies the conditions of PDP outlined in the work of Weymark (1981) and the convergence to leximin ordering is described in the work of Endriss (2018). Concretely, suppose that there are 3 parties. The party with the largest, mid, and smallest U^i_t is weighted 1, ρ, and ρ^2, respectively. As ρ decreases from 1 to 0, the relative differences between the parties' weights increase. Minimizing the fair regret s_t

defined w.r.t. ρ would lead to more fairness and smaller U_t.

Remark 3. How would g_t compare with u_t when varying ρ? Note that as ρ increases, $\sum_{i=1}^{n} w_i$ increases and becomes n when $\rho = 1$. For a better comparison, we should use normalized weights $w_i / \sum_{i=1}^{n} w_i$ to compute the normalized g_t. For a fixed $(f(\mathbf{x}_t^i) + U_{t-1}^i)_{i\in[n]}$, when ρ decreases, the normalized weights of the parties with smaller individual t-CUs increase, hence decreasing the normalized g_t to be less than u_t/n. The normalized g_t and S_T (i.e., computed using the normalized weights) will be used in Sec. 5.

Remark 4. There are other ways to set $\mathbf{w}$ to control the trade-off between efficiency vs. fairness. For example, we can set $w_i = \rho(n-i)+1$ for all $i \in [n]$ with $\rho > 1$ s.t. the weight w_i decreases linearly in i, like in the original Gini weights (see paragraph after (4)). However, it is not used as the preference of utility vectors induced by g_t converges to Leximin ordering more slowly, as compared to the exponentially decreasing weights in Proposition 1.

4. Collaborative BO with Fair Regret

We will now design a collaborative BO algorithm whose convergence rate can be theoretically guaranteed by bounding its *fair* (cumulative) regret s'_t (3) ($S'_T \triangleq \sum_{t=1}^{T} s'_t$).[5] To achieve this, we propose a variant of the DB-GP-UCB algorithm (Sec. 2.1) which jointly selects n input queries $\mathbf{X}_t = (\mathbf{x}_t^i)_{i\in[n]}$ to be assigned to all n parties in each iteration t by maximizing the following acquisition function that accounts for fairness via G2SF:

$$\mathbf{X}_t \triangleq \arg\max_{\mathbf{X}_t \in \mathcal{X}^n} \sum_{i=1}^{n} w_i \phi_i((\lambda_t^i + \mu_{\mathbf{x}_t^i|D_{1:t-1}})_{i\in[n]}) + \sqrt{\alpha_t \mathbb{I}(\mathbf{f}_{\mathcal{X}}; \mathbf{y}_t \mid D_{1:t-1})} \quad (5)$$

where ϕ is the same sorting operator defined previously in (4), ϕ_i is the i-th element of the sorted vector returned by ϕ, $\lambda_t^i \triangleq \sum_{t'=1}^{t-1} y_{t'}^i$ is a *realized/observed* variant of the individual $(t-1)$-CU (i.e., U_{t-1}^i) of each party i, and α_t is still a parameter that is set to trade off between exploitation vs. exploration. The first (exploitation) term in (5) is an *ordered weighted average* of $\lambda_t^i + \mu_{\mathbf{x}_t^i|D_{1:t-1}}$ for all $i \in [n]$ and thus satisfies both monotonicity (E1) and the PDP.

In each iteration t, our collaborative BO algorithm (5) selects n input queries $\mathbf{X}_t$ and assigns each input query $\mathbf{x}_t^i$ in $\mathbf{X}_t$ to party i for evaluating the objective function f. Such a selection has to trade off between (a) sampling near to a likely maximizer (i.e., with a large posterior mean $\mu_{\mathbf{x}_t^i|D_{1:t-1}}$), (b) sampling inputs that can yield large information gain $\mathbb{I}(\mathbf{f}_{\mathcal{X}}; \mathbf{y}_t \mid D_{1:t-1})$ to improve the belief of $f_{\mathcal{X}}$ (i.e., exploration), and (c) balancing the expected individual t-CUs between the parties by correcting past observed unfairness in λ_t^i. To achieve (c), the parties with smaller λ_t^i should exploit inputs with larger posterior mean while the parties with larger λ_t^i may be assigned to sample inputs with large information gain. Specifically, given any $\mathbf{X}_t$, the first term in (5) is maximized when the party with the k-th smallest λ_t^i is assigned to evaluate f at $\mathbf{x}$ with the k-th largest $\mu_{\mathbf{x}|D_{1:t-1}}$ due to the PDP property of function W (see Appendix B.2). So, our algorithm requires both ordered weighted averaging and λ_t^i to achieve fairness. Note that without the second exploration term, maximizing our acquisition function in (5) is equivalent to maximizing each party's expected utility separately. Though G2SF is only applied to the first exploitation term in (5), fairness affects the second exploration term and the exploitation-exploration trade-off through α_t which will be defined in Theorem 1.

Remark 5. One may be inspired by the acquisition function in (5) to consider an alternative two-step batch BO algorithm to incorporate fairness: (a) Select $\mathbf{X}_t$ using the DB-GP-UCB algorithm (1), and (b) permute the order of $\mathbf{x}_t^i$ in $\mathbf{X}_t$ s.t. $\mathbf{x}_t^i$ with a larger expected utility $\mu_{\mathbf{x}_t^i|D_{1:t-1}}$ is assigned to (or paired with) a smaller λ_t^i. However, this heuristic can only partially alleviate some fairness concerns as it will ignore the *numerical* values of each λ_t^i when selecting $\mathbf{X}_t$ and may be less aggressive at correcting for past unfairness. More importantly, it does not allow us to control the trade-off between efficiency vs. fairness.

4.1. Regret Bound

The convergence rate of our collaborative BO algorithm (5) can be theoretically guaranteed via a probabilistic bound on its fair cumulative regret S'_T:

Theorem 1. *Let $\delta \in (0, 1)$, $\gamma_T \triangleq \max_{\mathbf{X}_{1:T}} \mathbb{I}[\mathbf{f}_{\mathcal{X}}; \mathbf{y}_{1:T}]$, C be a constant with $C \geq \mathbb{I}[f_{\mathbf{x}}; (y_t^{i'})_{i'\in[i]}^{\top} \mid D_{1:t-1}]$ for all $i \in [n-1]$, $\mathbf{x} \in \mathcal{X}$, and $t = 1, \ldots, T$, $C_1 \triangleq 4/\log(1+\sigma^{-2})$, and $\alpha_t \triangleq C_1(\sum_{i=1}^{n} w_i^2)\exp(2C)\log(|\mathcal{X}|t^2\pi^2/(6\delta))$. Then, for the modified fair cumulative regret S'_T incurred by our collaborative BO algorithm (5), $S'_T = \sum_{t=1}^{T} s'_t \leq 2(T\alpha_T\gamma_T)^{1/2}$ holds with probability of at least $1-\delta$.*

Its proof and more details about the constants can be found in Appendix B.3. With the exception of α_t, the constants are the same as the ones defined in the work of Daxberger & Low (2017). Theorem 1 implies that $\lim_{T\to\infty} S'_T/T = 0$ and our algorithm can eventually converge to a global maximum, hence achieving asymptotic no-regret performance. That is, there exists an iteration t where $s'_t = 0$ and all parties evaluate f at $\mathbf{x}^*$ with no inefficiency and unfairness introduced. By bounding s'_t, we bound the inefficiency and unfairness (i.e., adjusted based on historical observed utilities) introduced in iteration t, i.e., we rule out some inefficient and less fair assignments, as described in Fig. 1.

Note that a smaller ρ for our collaborative BO algorithm

[5] BO algorithms usually cannot minimize R_T or S'_T directly.

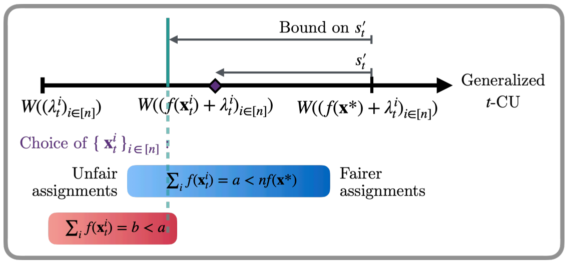


Figure 1. Overview of our notion of instantaneous *fair regret*: s'_t considers λ^i_t (i.e., realized/observed variant of individual $(t-1)$-CU which cannot be changed) and is the loss in the generalized t-CU g'_t from evaluating f at $\{\mathbf{x}^i_t\}_{i\in[n]}$ instead of the maximizer $\mathbf{x}^*$. As W captures both efficiency and fairness, when $s'_t > 0$, s'_t reflects the inefficiency from not evaluating $\mathbf{x}^*$ in iteration t and some unfairness. The red and blue boxes enclose assignments with the same full utility u_t of values a and b, respectively. A fairer assignment is of a darker shade and increases g'_t. By bounding s'_t, we rule out some assignments due to their inefficiency (almost the entire red box) or unfairness (pale blue).

would lead to relatively more input queries selected for exploration over exploitation due to two reasons: (a) Observe that α_t in Theorem 1 depends on the weights w_i's of G2SF. A smaller ρ will lead to a greater ratio of $\sqrt{\sum_{i=1}^n w_i^2}$ in the exploration term to $\sum_{i=1}^n w_i$ (i.e., total weight of the exploitation term). (b) Even after adjusting for the ratio, a smaller ρ would lead to a smaller exploitation term in (5) (see Remark 3 in Sec. 3.3). This increased exploration may cause the observed standard full cumulative regret R_T (and possibly S'_T) to increase. Furthermore, as the order of parties in the computation of g'_t may change across iterations, a smaller ρ (i.e., associated with more fairness) would reduce the efficiency of multiple parties.

5. Experiments and Discussion

This section empirically evaluates the performance and properties of our collaborative BO algorithm using a benchmark function: Hartmann-$6d$, and three real-world collaborative black-box optimization problems: (a) hyperparameter tuning of a *logistic regression* (LR) model with a mobile sensor dataset (Anguita et al., 2013), (b) hyperparameter tuning of a *convolutional neural network* (CNN) with *federated extended MNIST* (FEMNIST) dataset (Caldas et al., 2018), and (c) mobility demand hotspot discovery on a traffic dataset (Chen et al., 2013). The performance of our collaborative BO algorithm is compared with that of the *distributed batch GP-UCB* (DB-GP-UCB) algorithm (Daxberger & Low, 2017) which has been empirically shown to outperform several existing batch BO algorithms (Daxberger & Low, 2017; Kharkovskii et al., 2020) but, like them, does not consider fairness. Note that DB-GP-UCB is equivalent to our algorithm with $\rho = 1$, i.e., each party's expected individual t-CU has equal weight. We evaluate the performance of the tested algorithms in terms of the following metrics:

- Inefficiency, measured using full *cumulative regret* (CR) (Sec. 2.2) averaged across parties, i.e., R_T/n;
- Unfairness, measured using $U_t/n - g_t$ which is the difference between the averaged individual t-CU vs. the generalized t-CU. When computing g_t, we set[6] $\rho = 0.2$ and normalize the weights w_i s.t. $\sum_{i=1}^n w_i = 1$ for a fair comparison. So, $U_t/n - g_t$ is the generalized Gini absolute inequality indices (Weymark, 1981) and is an unfairness metric that satisfies PDP as g_t satisfies PDP (Proposition 1) and U_t/n is a constant if the precondition of PDP holds. Its minimum value is 0 when U^i_t for all $i \in [n]$ are equal (i.e., the fairest outcome);
- Unfairness and inefficiency, measured together using the normalized S_T by setting $\rho = 0.2$. For the real-world experiments, as $f(\mathbf{x}^i_t)$ is unknown, we approximate $f(\mathbf{x}^i_t)$ with y^t_i for all $i \in [n]$ and $t = 1, \ldots, T$ and so, $S_T = S'_T$.

We aim to show that our collaborative BO algorithm with $\rho < 1$ can improve fairness and observe the effect on efficiency. Following the previous GP-UCB works (Kandasamy et al., 2015; 2016) and our theoretical result (Theorem 1), we set $\alpha_t = c_1 d(\sum_{i=1}^n w_i^2)\log(c_2 t)$ in (5). We consider two settings: (i) *Fix* c_1: c_1 and c_2 are fixed across different ρ's, and (ii) *Vary* c_1: c_2 is fixed but c_1 varies for different ρ's s.t. the ratio of $\sqrt{\sum_{i=1}^n w_i^2}$ in the exploration term to $\sum_{i=1}^n w_i$ (i.e., total weight of the exploitation term) is a constant. Recall from the last paragraph of Sec. 4.1 that a smaller ρ will increase the exploration relative to exploitation and hence reduce the efficiency. Setting (ii) is used to elaborate and mitigate this effect. Details on the choices of c_1 and c_2 are shown in Appendix C.1.

For each experiment, we repeat 10 runs of BO with different random seeds and plot the standard error as the error bars. The objective function is modeled as a GP with $k_{\mathbf{x}\mathbf{x}'}$ chosen to be a squared exponential kernel. We describe the experimental setting of each tested objective function below:

Hartmann-$6d$ function. We consider $n = 3$ parties. The objective function f has $d = 6$ input dimensions in $[0, 1]^6$. In each run, 10 data points are randomly selected for each party as initialization (i.e., $T_0 = 10$ in Algorithm 1 of Appendix A).

Hyperparameter tuning of LR with mobile sensor dataset. We consider $n = 5$ parties and each party trains a LR model for activity recognition using sensor data gathered by 10 mobile sensors.[7] These parties collaborate to tune

[6] Note that this is distinct from ρ used in (5) which is varied in each experiment.

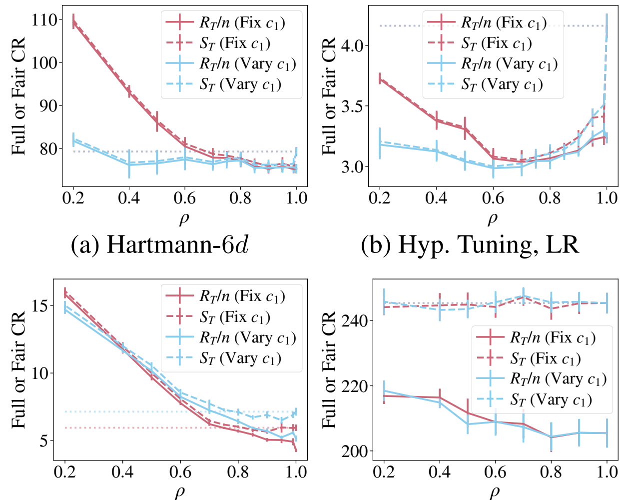
[7] Every party represents a company who owns the data of multiple users. They share a common objective function as the distributions of the companies' data pooled from multiple users are likely

three hyperparameters of a LR: batch size in $[20, 100]$, L2 regularization parameter and learning rate in $[10^{-5}, 1]$. The output of the objective function is the validation accuracy of the LR model. We set $T_0 = 2$. Every party is particular about its individual CU and the best u_t^i that it has evaluated as it would use the corresponding LR model for real-world activity recognition.

Hyperparameter tuning of CNN with FEMNIST. We consider $n = 10$ parties and each party trains a CNN for character recognition using handwritten digits from 10 authors. The parties collaborate to tune three hyperparameters: learning rate, learning rate decay, and regularization parameters in $[10^{-5}, 1]$. The output of the objective function is the validation accuracy of the CNN model and $T_0 = 2$.

Mobility demand hotspot discovery on traffic dataset. The traffic dataset includes 2506 input regions, each of which has a mobility demand.[8] We adopt the settings in (Chen et al., 2015; Kharkovskii et al., 2020) and consider $n = 8$ parties. Each party is a company that operates a fleet of MoD AVs and wants to discover the region with the highest mobility demand.[9] If an MoD AV picks up a passenger, the company deploys another MoD AV to take its place. In each BO iteration, the mediator would assign each company to evaluate the mobility demand at an input region that is connected/near to its current input region. This constraint makes it harder to achieve efficiency and fairness as an MoD AV cruising in low mobility demand regions cannot evaluate at known higher mobility demand regions that are too far away. The aim here is to show that our collaborative BO algorithm can improve fairness.[10]

Observations about the notions of regret. Fig. 2 shows results of the full CR R_T/n and fair CR S_T against different ρ's used in our collaborative BO algorithm (5) for each experiment. It can be observed that as ρ decreases, the inefficiency metric R_T/n generally increases for both fixed and varying c_1. This is expected since with a smaller ρ, the relative weight of the exploration term in (5) is larger and the exploitation term is smaller (i.e., discussed in Sec. 4.1 and Proposition 1), which reduces efficiency. However, in Fig. 2b, R_T/n decreases when ρ decreases from 1 to around 0.7. This may be due to the objective function of the LR being easy to maximize. For example, there are multiple hyperparameters that can achieve similar competitive validation accuracies for LR. Therefore, increasing the exploration (i.e., decreasing ρ) may lead to earlier discovery of inputs with small regret and thus improve efficiency.



(a) Hartmann-6d (b) Hyp. Tuning, LR

(c) Hyp. Tuning, CNN (d) Traffic

Figure 2. Full CR R_T/n and normalized fair CR S_T averaged over 10 BO runs incurred by the tested algorithms using different ρ's in (5). The dotted horizontal lines represent the results of $\rho = 1$ (i.e., DB-GP-UCB).

Moreover, as ρ decreases from 1 to 0.2, Fig. 2 shows that the fair CR S_T, which measures both inefficiency and unfairness, first decreases and then increases. In Fig. 2, we plot the fair CR S_T of DB-GP-UCB (equivalent to ours with $\rho = 1$) as a dotted line to aid comparison with S_T of other ρ's. The decrease of S_T is due to the improved fairness (and possibly efficiency) while the subsequent increase of S_T is due to larger inefficiency having a dominant effect. For each experiment, since DB-GP-UCB ($\rho = 1$) does not consider fairness, its S_T may be larger than those with smaller ρ's even though its R_T/n (when $\rho = 1$) is smaller. Furthermore, it can be observed that when ρ is small, the blue lines (Vary c_1) usually lie below the red lines (Fix c_1), which shows that varying c_1 can reduce the inefficiency significantly. In Figs. 2b-c, the gap between S_T and R_T/n, which reflects unfairness, is the largest when $\rho = 1$. This is expected as when $\rho = 1$, the mediator may unintentionally assign a party to evaluate at inputs with smaller function values multiple times. In Appendix C.2, we plot more graphs including R_t/n vs. iteration t for various ρ's; the observations are similar to the above.

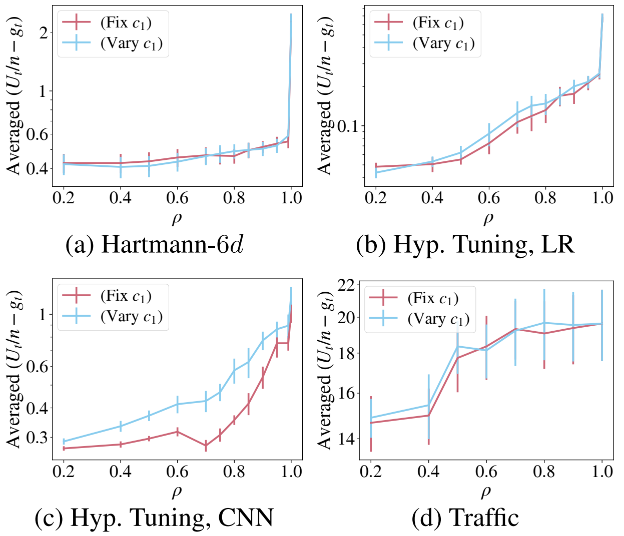
Observations about fairness. To evaluate the fairness achieved by the tested algorithms, Fig. 3 shows results of the unfairness metric $U_t/n - g_t$ averaged over iterations $t = 1, \ldots, T$ against different ρ's used in our collaborative BO algorithm (5). It can be observed that for all the experiments, the averaged $(U_t/n - g_t)$ decreases as ρ decreases. This observation holds for various choices of fixed

to be similar.

[8] Appendix C.1 visualizes the mobility demand distribution.

[9] An MoD AV visiting a higher mobility demand region is more likely to pick up passengers, earn fares, and collect more useful information.

[10] The efficiency of our collaborative BO algorithm in this experiment can be improved via a non-myopic approach (Kharkovskii et al., 2020), or relaxing the locality constraints and adopting discrete optimization techniques to maximize the acquisition function. However, this is non-trivial and not the focus of this work.



(a) Hartmann-6d (b) Hyp. Tuning, LR (c) Hyp. Tuning, CNN (d) Traffic

Figure 3. Averaged $(U_t/n - g_t)$ incurred by the tested algorithms using different ρ's in (5) for various objective functions.

c_1 and even for varying c_1. Combining this observation with that about R_T/n, we suggest to use varying c_1 in the real-world applications for achieving more fairness and efficiency. However, one may ask whether an increase in $U_t/n - g_t$ is due to more unfairness or the increase of the U_t/n alone. To answer this question, we plot $U_t/n - g_t$ vs. U_t/n for different ρ's in Appendix C.3. It shows that even for the same U_t/n, $U_t/n - g_t$ generally decreases when ρ decreases, which indicates that our collaborative BO algorithm with a smaller ρ can achieve fairer outcomes.

Individual rationality. In CGT, a desirable property of a collaboration is *individual rationality*, which states that each party should not be worse off via collaborating than working alone (Chalkiadakis et al., 2011). In our collaborative BO setting, it is *individually rational* for any party to participate in the collaboration if it can obtain a better estimate of $\mathbf{x}^*$ for the same CR or, alternatively, smaller CR for the same estimate of $\mathbf{x}^*$ compared to performing BO alone. In Appendix C.5, we show that for some ρ's, our collaborative BO algorithm would satisfy this property by assuming that parties would perform GP-UCB (Srinivas et al., 2010) on their own. More experimental results on the simple regret, collaborative BO with $n = 50$ parties, and some detailed analysis of λ_t^i are given in Appendix C.

6. Related Work

To the best of our knowledge, there is no existing work on collaborative fairness in BO. The closely related works are *multi-armed bandit* (MAB) with various fairness considerations and batch BO, as summarized in Table 1. Our work builds upon batch BO but additionally minimizes the differences between the individual CUs of all parties to encourage collaboration. The MAB works consider significantly different notions of fairness. Only the work of Hossain et al. (2020) has studied fairness across parties but its setting differs as every party receives a (different) utility whenever an arm is pulled. The *incentivizing exploration* (IE) works are not designed to ensure fairness across parties. Instead, they tackle a separate problem of incentivizing each self-interested party to adopt the mediator's recommendation and explore arms with smaller utility, which benefits the group but may not benefit itself. Another key difference is that in most IE applications, parties are less keen on exploration as they are not intent on finding the maximizer $\mathbf{x}^*$ and do not participate in multiple rounds. Nevertheless, designing a *fair* and *incentive-compatible* (Mansour et al., 2020) collaborative BO algorithm that does not rely on external incentives (e.g., monetary payment) is an important but nontrivial future work.

Table 1. Summary of related works.

Description	References
Batch BO. Select the input of each party one at a time	(Alvi et al., 2019; Azimi et al., 2010; Contal et al., 2013; Desautels et al., 2014; González et al., 2016)
Batch BO. Optimize the batch of inputs jointly	(Chevalier & Ginsbourger, 2013; Daxberger & Low, 2017; Shah & Ghahramani, 2015; Wu & Frazier, 2016)
MAB. Fairness across arms	(Liu et al., 2017; Patil et al., 2020)
MAB. Fairness between objectives	(Busa-Fekete et al., 2017)
MAB. Fairness across parties	(Hossain et al., 2020)
MAB. Incentivizing exploration	(Frazier et al., 2014; Mansour et al., 2020)

7. Conclusion

This paper describes the first BO algorithm that considers collaborative fairness. We propose new notions of fair (cumulative) regret and a collaborative BO algorithm whose convergence rate can be theoretically guaranteed by bounding the new fair regret. By controlling parameter ρ, the parties can select a set of weights for G2SF to trade off between efficiency vs. fairness. This is empirically demonstrated using a benchmark function and three real-world experiments. With this collaborative BO algorithm that considers each party's historical utilities, a mediator can ensure fairness *across* iterations while *greedily* selecting input queries for the current iteration. By considering fairness in every iteration up to T, the differences between the (historical) CUs of all parties are reduced and can thus be corrected without significantly hurting efficiency in the current iteration.

Acknowledgements

This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (Award No: AISG2-RP-2020-018). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore. R. H. L. Sim is also supported by the Singapore-MIT Alliance for Research and Technology (SMART) Ph.D. Fellowship.

References

Alvi, A. S., Ru, B., Calliess, J., Roberts, S. J., and Osborne, M. A. Asynchronous batch Bayesian optimisation with improved local penalisation. In *Proc. ICML*, 2019.

Anguita, D., Ghio, A., Oneto, L., Parra, X., and Reyes-Ortiz, J. L. A public domain dataset for human activity recognition using smartphones. In *Proc. ESANN*, 2013.

Azimi, J., Fern, A., and Fern, X. Z. Batch Bayesian optimization via simulation matching. In *Proc. NeurIPS*, pp. 109–117, 2010.

Busa-Fekete, R., Szörényi, B., Weng, P., and Mannor, S. Multi-objective bandits: Optimizing the generalized Gini index. In *Proc. ICML*, pp. 625–634, 2017.

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečnỳ, J., McMahan, H. B., Smith, V., and Talwalkar, A. LEAF: A benchmark for federated settings. arXiv:1812.01097, 2018.

Chalkiadakis, G., Elkind, E., and Wooldridge, M. Computational aspects of cooperative game theory. In Brachman, R. J., Cohen, W. W., and Dietterich, T. G. (eds.), *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan & Claypool Publishers, 2011.

Chen, J., Low, K. H., and Tan, C. K.-Y. Gaussian process-based decentralized data fusion and active sensing for mobility-on-demand system. In *Proc. RSS*, 2013.

Chen, J., Low, K. H., Jaillet, P., and Yao, Y. Gaussian process decentralized data fusion and active sensing for spatiotemporal traffic modeling and prediction in mobility-on-demand systems. *IEEE Trans. Autom. Sci. Eng.*, 12: 901–921, 2015.

Cheng, F., Ma, Y., Uzzi, B., and Loscalzo, J. Importance of scientific collaboration in contemporary drug discovery and development: A detailed network analysis. *BMC Biology*, 18(138), 2020.

Chevalier, C. and Ginsbourger, D. Fast computation of the multi-points expected improvement with applications in batch selection. In *Proc. LION*, pp. 59–69, 2013.

Contal, E., Buffoni, D., Robicquet, A., and Vayatis, N. Parallel Gaussian process optimization with upper confidence bound and pure exploration. In *Proc. ECML/PKDD*, pp. 225–240, 2013.

Dai, Z., Yu, H., Low, B. K. H., and Jaillet, P. Bayesian optimization meets Bayesian optimal stopping. In *Proc. ICML*, pp. 1496–1506, 2019.

Dai, Z., Chen, Y., Low, B. K. H., Jaillet, P., and Ho, T.-H. R2-B2: Recursive reasoning-based Bayesian optimization for no-regret learning in games. In *Proc. ICML*, 2020a.

Dai, Z., Low, B. K. H., and Jaillet, P. Federated Bayesian optimization via Thompson sampling. In *Proc. NeurIPS*, pp. 9687–9699, 2020b.

Dalton, H. The measurement of the inequality of incomes. *The Economic Journal*, 30(119):348–361, 1920.

Daxberger, E. A. and Low, B. K. H. Distributed batch Gaussian process optimization. In *Proc. ICML*, pp. 951–960, 2017.

Desautels, T., Krause, A., and Burdick, J. W. Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization. *JMLR*, 15:4053–4103, 2014.

Endriss, U. Lecture notes on fair division. arXiv:1806.04234, 2018.

Fitch, D. Opinion: The value of collaboration in Ag technology, 2018. URL https://www.precisionag.com/digital-farming/.

Frazier, P., Kempe, D., Kleinberg, J., and Kleinberg, R. Incentivizing exploration. In *Proc. EC*, pp. 5–22, 2014.

González, J., Dai, Z., Hennig, P., and Lawrence, N. D. Batch Bayesian optimization via local penalization. In *Proc. AISTATS*, 2016.

Griffiths, R. and Hernández-Lobato, J. M. Constrained Bayesian optimization for automatic chemical design using variational autoencoders. *Chemical Science*, 11(2): 577–586, 2020.

Hoang, T. N., Hoang, Q. M., and Low, K. H. A unifying framework of anytime sparse Gaussian process regression models with stochastic variational inference for big data. In *Proc. ICML*, pp. 569–578, 2015.

Hoang, T. N., Hoang, Q. M., and Low, K. H. A distributed variational inference framework for unifying parallel sparse Gaussian process regression models. In *Proc. ICML*, pp. 382–391, 2016.

Hoang, T. N., Hoang, Q. M., Ouyang, R., and Low, K. H. Decentralized high-dimensional Bayesian optimization with factor graphs. In *Proc. AAAI*, pp. 3231–3238, 2018.

Hossain, S., Micha, E., and Shah, N. Fair algorithms for multi-agent multi-armed bandits. arXiv:2007.06699, 2020.

Kandasamy, K., Schneider, J., and Póczos, B. High dimensional Bayesian optimisation and bandits via additive models. In *Proc. ICML*, pp. 295–304, 2015.

Kandasamy, K., Dasarathy, G., Oliva, J., Schneider, J., and Póczos, B. Gaussian process optimisation with multi-fidelity evaluations. In *Proc. NeurIPS*, 2016.

Kharkovskii, D., Ling, C. K., and Low, K. H. Nonmyopic Gaussian process optimization with macro-actions. In *Proc. AISTATS*, pp. 4593–4604, 2020.

Ling, C. K., Low, K. H., and Jaillet, P. Gaussian process planning with Lipschitz continuous reward functions: Towards unifying Bayesian optimization, active learning, and beyond. In *Proc. AAAI*, pp. 1860–1866, 2016.

Liu, Y., Radanovic, G., Dimitrakakis, C., Mandal, D., and Parkes, D. Calibrated fairness in bandits. In *Proc. KDD Workshop on Fairness, Accountability, and Transparency in Machine Learning*, 2017.

Mansour, Y., Slivkins, A., and Syrgkanis, V. Bayesian incentive-compatible bandit exploration. *Operations Research*, 68(4):1132–1161, 2020.

Patil, V., Ghalme, G., Nair, V., and Narahari, Y. Achieving fairness in the stochastic multi-armed bandit problem. In *Proc. AAAI*, pp. 5379–5386, 2020.

Perrone, V., Donini, M., Kenthapadi, K., and Archambeau, C. Fair Bayesian optimization. arXiv:2006.05109, 2020.

Pigou, A. C. *Wealth and Welfare*. Macmillan and Co., London, 1912.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Shah, A. and Ghahramani, Z. Parallel predictive entropy search for batch global optimization of expensive objective functions. In *Proc. NeurIPS*, pp. 3312–3320, 2015.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104 (1):148–175, 2015.

Sim, R. H. L., Zhang, Y., Chan, M. C., and Low, B. K. H. Collaborative machine learning with incentive-aware model rewards. In *Proc. ICML*, pp. 8927–8936, 2020.

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proc. ICML*, pp. 1015–1022, 2010.

Weymark, J. A. Generalized Gini inequality indices. *Mathematical Social Sciences*, 1(4):409–430, 1981.

Wu, J. and Frazier, P. The parallel knowledge gradient method for batch Bayesian optimization. In *Proc. NeurIPS*, pp. 3126–3134, 2016.

Zhang, Y., Dai, Z., and Low, B. K. H. Bayesian optimization with binary auxiliary information. In *Proc. UAI*, pp. 1222–1232, 2019.

Zhang, Y., Apley, D. W., and Chen, W. Bayesian optimization for materials design with mixed quantitative and qualitative variables. *Sci. Rep.*, 10(4924), 2020.