big data and cognitive computing

MDPI

Article

Resource Allocation via Bayesian Optimization in Wasserstein Spaces vs. Semi-Bandit Feedback

Antonio Candelieri [1], Francesco Archetti [2,*], Iman Seyedi [2,*] and Andrea Ponti [1]

[1] Department of Economics Management and Statistics, University of Milano-Bicocca, 20126 Milan, Italy; antonio.candelieri@unimib.it (A.C.); a.ponti5@campus.unimib.it (A.P.)
[2] Department of Computer Science Systems and Communication, University of Milano-Bicocca, 20126 Milan, Italy
* Correspondence: francesco.archetti@unimib.it (F.A.); seyediman.seyedi@unimib.it (I.S.)

Abstract

Sequential resource allocation has long been a central problem in operations research, yet ongoing technological developments, particularly in cloud and high-performance computing and in multi-channel marketing, are giving rise to new structural constraints that classical methods were not designed to handle. Semi-Bandit Feedback (SBF) has emerged as the dominant framework for these modern settings. This paper introduces an alternative that recasts the allocation problem within the Bayesian Optimization (BO) paradigm. All three proposed BO algorithms consistently outperform SBF, with BORAwse showing a particularly clear advantage under time-varying budget settings, while CBO achieves comparable rewards under constant budget conditions. The core methodological contribution is a reformulation in which each candidate allocation is represented as a discrete probability distribution over the available options, making the probability simplex the natural search domain. Grounding the search in this space calls for a geometry that respects the structure of distributions: we adopt the optimal transport (Wasserstein) distance, which allows both the Gaussian process surrogate and the acquisition function to be extended as functionals over the simplex. A further practical advantage of the proposed method is its applicability to problem instances where SBF cannot be used without modification. The approach is evaluated on two case studies: the benchmark computing-resource allocation scenario from the original SBF paper, and a budget allocation problem across marketing channels.

Keywords: resource allocation; semi-bandit feedback; Bayesian optimization; Wasserstein distance; optimal transport

Academic Editor: Fabrizio Messina

Received: 27 March 2026
Revised: 10 June 2026
Accepted: 22 June 2026
Published: 25 June 2026

1. Introduction

The reference problem considered in this paper is the optimal sequential resource allocation. Assume that, at each time step t, a decision-maker has to distribute a given budget $\bar{b}^t$ over m different options to maximize the cumulative reward of her/his decisions in T time steps. Formally, she/he solves:

$$\max_{x(t)\in R_+^m} \mathbb{E}\left[\sum_{t=1}^{T} f(x^t)\right] \quad s.t. \sum_{i=1}^{m} x_i^t \le \bar{b}^t \tag{1}$$

where $f(x^t)$ is the immediate reward associated with the decision x^t made at time t. In some cases, the constraint in (1) is considered as an equality instead of an inequality (i.e., all the budget must be used), as done later in this paper. Optimal (sequential) resource allocation is a well-known problem in operations research, with traditional examples such as inventory management, portfolio allocation, etc. [1–4]. Some approaches are related to stochastic optimization (e.g., multi-period and multi-stage stochastic optimization [5–7]), but the most relevant literature for this paper is from the multi-armed bandit (MAB) [8,9], specifically the semi-bandit feedback (SBF) [10–14]. SBF has recently gained renewed interest, according to the technological advances and widespread usage of cloud and/or high-performance computing facilities [15–19], where it is critical to optimally allocate computational resources (i.e., the budget) to different jobs (i.e., options, arms) to maximize the number of completed jobs over time. Analogously to MAB, SBF also requires that every decision (x^t) has to be made by balancing between exploitation (i.e., making the best decision according to the current knowledge about the problem) and exploration (i.e., making a decision with the aim to reduce uncertainty instead of settling for the currently known best one). While MAB consists of selecting, at each step t, just one option (aka arm) among the m possible, SBF allows for simultaneously choosing multiple arms, by allocating a different amount $x_i^t \ge 0$ of the available budget on each of them. The reward will depend on the amount of budget allocated to all the arms altogether. Indeed, it is important to remark the difference with Combinatorial MAB (CMAB), where more than one arm can be pulled at each time step (like in SBF), but decision variables, (x^t), are binary instead of numeric [18], as well as with best arm identification (BAI), where the goal is to identify just a single arm providing the highest reward [20].

In the original formulation of SBF, proposed in [21] and successively improved in [22], the options are computer processes (a.k.a. jobs), the budget is constant over time, and rescaled to 1 so that the constraint becomes $\sum_{i=1}^{m} x_i^t \le 1$ and, consequently, x^t represents the allocation, in percentage, of the resources to the m different computer processes at time step t. Accordingly, the immediate reward is the number of completed processes, whose individual probabilities of completion are given by Bernoulli distributions, $\mathcal{B}(x_i^t/v_i) := min\{1, x_i^t/v_i\}$, with v_i unknown cutoff parameters: the smaller v_i, the easier the completion of the associated job.

Thus, the immediate reward is $f(x^t) = \sum_{i=1}^{m} \mathcal{B}(x_i^t/v_i)$. Since values $v_1, \ldots, v_m$ are unknown, the immediate reward is a black box. It is also important to clarify that, at the end of every time step, the resources are replenished, and all jobs are reset regardless of whether or not they completed in the previous time step (i.e., there is no inter-temporality between consecutive decisions). The aim is to learn, as fast as possible, the values $v_1, \ldots, v_m$, to choose (x^t) maximizing $\mathbb{E}\sum_{t=1}^{T}\sum_{i=1}^{m} \mathcal{B}(x_i^t/v_i)$, $with \sum_{i=1}^{m} x_i^t \le 1$.

Bayesian Optimization (BO) offers a complementary learning-and-optimization perspective. It constructs a probabilistic surrogate of the objective and uses it to guide a sequence of queries toward the global optimum, making it particularly well-suited for functions that are expensive to evaluate, non-convex, or corrupted by noise [23–26].

The GP-based formulation of BO (GP-BO) has established itself as the dominant paradigm for tackling black-box, computationally expensive optimization tasks, demonstrating broad applicability across engineering and machine learning domains. Despite this success, GP-BO suffers from a well-known scalability bottleneck: its performance degrades substantially when the number of decision variables exceeds roughly 15–20 [25,27].

One productive response to this limitation has been to recognize that many practical problems possess a lower effective dimensionality than their nominal parametrization suggests, motivating methods that embed the search into a more compact representation.

The approach taken in this paper exploits a different geometric observation: since the resource allocation problem is inherently about distributing a budget, the natural search domain is not a box-bounded Euclidean space but the probability simplex, whose elements are discrete probability distributions. We therefore reformulate the optimization entirely within the Wasserstein space—a nonlinear manifold metrized by the optimal transport distance—so that both the GP surrogate and the acquisition function become functionals defined over distributions rather than over points. Thus, the optimization of the acquisition functional can be carried out via two alternative strategies: gradient-based updates through automatic differentiation constrained over the probability simplex or a proximal-point scheme driven by the Wasserstein gradient flow. As shown in our experiments, this geometric reformulation yields consistent gains over standard BO, and those gains widen as the number of arms—and hence the dimension of the allocation problem—grows. Moreover, a simple gradient-based method constrained to the probability simplex is sufficient to obtain better results than standard BO, without the need for implementing a more computationally expensive Wasserstein gradient flow schema.

Like MAB, BO operates sequentially and must continually balance exploitation against exploration; however, the two frameworks differ fundamentally in the nature of their search spaces—at least in their classical formulations. In MAB, each decision amounts to choosing a single arm from a finite, discrete set of m options, whereas in BO each decision corresponds to selecting a point from a continuous, box-bounded domain. To make this precise, consider the following global optimization problem:

$$\max_{x \in \Omega \subset \mathbb{R}^m} f(x) \tag{2}$$

At each step t, the gap between the true optimum $f(x^*)$ and the value obtained by the current decision $f(x^t)$ defines the instantaneous regret $r^t = f(x^*) - f(x^t)$, and its sum over the horizon, $R^T = \sum_{t=1}^{T} r^t$, gives the cumulative regret. A fundamental theoretical result established in [28] is that, under appropriate regularity conditions, BO achieves sub-linear cumulative regret, i.e., $\lim_{T \to \infty} \frac{R^T}{T} = 0$, guaranteeing that the average per-step loss vanishes as the number of iterations grows.

The structural parallels between BO and MAB have motivated a rich line of work that transfers bandit policies into the BO setting. In the MAB problem [8,9], an agent facing a fixed collection of arms must repeatedly choose which one to pull, receiving each time a stochastic payoff sampled from a distribution that is specific to that arm and unknown in advance. Since no arm's reward distribution is revealed without pulling it, the agent must decide at every round how to allocate its limited query budget—committing to arms that appear rewarding based on past observations, or investing in arms whose behavior remains uncertain. The overarching objective is to find a sequential selection strategy that maximizes total accumulated reward across all rounds.

Since the payoff characteristics of each arm are initially hidden, they can only be inferred through repeated interaction with the system. Gaining knowledge about any arm requires actively querying it and recording what it returns. The agent is faced with an exploration/exploitation dilemma: repeatedly forced to decide whether to select an arm believed to have a high expected reward or to sample an uncertain arm to better understand its reward distribution.

A useful bridge to continuous optimization is to treat the decision variable x as an arm index in an uncountably infinite collection, where querying f at any point x yields a noisy observation drawn from some local distribution.

A key assumption underlying the classical MAB formulation is that the reward of each arm is statistically independent of all others, given the chosen action. Under strict independence, pulling one arm yields no insight whatsoever into the returns of any other arm. Making the infinite arms bandit hopeless problem, instead, we must assume that rewards, analogously to observations in BO, are correlated so that each observation can potentially inform us about the rewards of every other arm.

At its core, MAB captures the fundamental challenge of sequential decision-making under uncertainty in online learning settings. In this framework, each of the n available options—indexed as M_i, $i = 1, \ldots, n$—generates a random payoff $r \sim P(r_i, \theta_i)$ governed by an unknown parameter θ_i specific to that option.

Given a finite budget of N interactions, the learner's goal is to determine the mean return $\bar{r}_i$ for each option M_i, yet this can only be achieved by committing a query to M_i and recording its realized output r_i.

MAB is often used to model the A/B testing problem and recommender systems, in which we are running an online advertisement system, and we can choose, dynamically, among different advertisements (e.g., banners) on the basis of their profitability.

Each query directed at a machine serves a dual purpose [23]:

- Increase one's knowledge.
- Increase one's reward.

Learning policies in MAB face the exploration versus exploitation dilemma, i.e., the search for a balance between exploring the environment to find profitable actions, while taking the empirically best action as often as possible [29].

At each round t, the learner selects an action $a_t \in (1, \ldots, n)$ corresponding to a particular machine; the resulting outcome $r_t \in \mathbb{R}$ has mean $\bar{y}$. The accumulated interaction record is then mapped to the next action through a decision rule:

$$\pi : (a_1, r_1), \ldots, (a_{t-1}, r_{t-1}) \rightarrow a_t \tag{3}$$

When the arm set is made continuous, this bandit perspective extends naturally to global optimization: the role of the action is played by the query point x, the probabilistic surrogate serves as the expected reward model, and each function evaluation f(x) constitutes the observed payoff.

From this viewpoint, BO can equally be understood as an active learning procedure: unlike standard supervised learning, where the training set is fixed a priori, the optimizer itself determines, step by step, which input locations x are worth evaluating.

We are concerned with GP-based optimization in the MAB setting, where the reward is sampled from a GP distribution. A desirable asymptotic property of an algorithm is to be no-regret: $\lim_{T \to \infty} \frac{R_T}{T} = 0$. Note that neither r_t nor R_T are ever revealed to the algorithm.

Quantitatively, controlling $\frac{R_T}{T}$ is equivalent to bounding the gap between the best value found within T evaluations, $\max_{x_t \leq T} f(x_t)$, and the true optimum $f(x^*)$, thereby establishing convergence rates for GP-based BO.

Among acquisition functions, GP Upper Confidence Bound (GP-UCB) stands out as the policy that achieves no-regret guarantees for the MAB, and does so by solving at each step:

$$x_{t+1} = \arg\max_{x \in X} \mu_t(x) + \xi_t \sigma_t(x) \tag{4}$$

where $\mu_t(x)$ and $\sigma_t(x)$ are the mean and standard deviation of the GP at step t.

This means that, if we could remove the constraint in (1) and impose $x^t \in \Omega, \forall\, t = \{1, \ldots, T\}$, then solving (2) via BO would lead to a solution for the unconstrained version of the problem (1). Consequently, a simple idea would be to address the problem (1) via Constrained BO (CBO) [30–32], with $\bar{b}^t$ potentially changing at each time step t.

https://doi.org/10.3390/bdcc10070206

Although possible, we are going to demonstrate that the CBO approach is not so efficient. The key difficulty is that resource allocation vectors are not arbitrary points in a box-bounded Euclidean space: they are constrained to lie on the probability simplex, a non-Euclidean manifold whose geometry is fundamentally different from $\mathbb{R}^m$. Standard BO operates in Euclidean spaces and imposes simplex constraints only as post hoc projections, losing the intrinsic geometric structure of the problem. To address this, we propose a novel method that performs BO natively over the $m-1$ dimensional probability simplex, Δ_{m-1}, equipped with the Wasserstein distance as its metric. This allows the GP surrogate and acquisition functional to be defined in a space that is geometrically consistent with the allocation problem itself, rather than being an approximation of it.

The initial formulation of the optimal resource allocation problem [21,22] is extended to the case of budget changing over time, that is $\sum_{i=1:m} x_i^{(t)} = \bar{b}^{(t)}$. The first aspect is a requirement emerging from real-life applications, where the budget $\bar{b}^t$ cannot be directly decided by the decision-maker, but it comes from other managing considerations.

In comparison with the previous conference paper [33], the experimental setting of the manuscript submitted to BDCC has been largely extended with the aim of addressing a deeper analysis of the performances with respect to different problem scales. Moreover, discussions and conclusions have also been significantly extended and improved, reporting pros and cons which could not be properly investigated and detailed in a conference paper. The deeper analysis we have provided in this manuscript will allow BDCC's readers to fully understand and reuse our approach as well as our results.

The paper is organized as follows. Section 2 reviews the related literature, covering prior work on resource allocation, semi-bandit feedback, Bayesian Optimization, and optimal transport. Section 3 provides the methodological background, introducing the optimal resource allocation problem via SBF and the foundations of Gaussian process-based Bayesian Optimization. Section 4 presents the Wasserstein distance and optimal transport framework, including entropic regularization, the Wasserstein gradient flow and proximal algorithm, and the positive definiteness of kernels in the Wasserstein space. Section 5 describes the three BO algorithms proposed for optimal resource allocation: CBO, BORAse, and BORAwse. Section 6 reports the experimental results on two real-life case studies. Section 7 concludes the paper and outlines directions for future work.

2. Related Works

The broader landscape of Optimal Resource Allocation is mapped in [34], which, despite its age, remains a useful reference for understanding the range of application domains and the variety of solution approaches that have been proposed. Within the more specific thread of SBF-based methods, a substantial body of recent work has accumulated [11–14,18], with [22,35,36] being the most directly relevant to the sequential allocation setting considered here. The BO literature most pertinent to this paper concerns its constrained variant (CBO), covered in [30–32]. On the bandit side, Zuo and Joe-Wong [37] tackle sequential allocation through a combinatorial multi-armed bandit (CMAB) lens, developing two distinct algorithms: CUCB-DRA, designed for settings where the budget takes discrete values, and CUCB-CRA, which handles continuous budget spaces through a fixed discretization scheme. CUCB-DRA achieves logarithmic regret under semi-bandit feedback, while CUCB-CRA handles continuous action spaces via fixed discretization. Kasy and Teytelboym [38] study adaptive matching of resources to participants under uncertainty, proposing a Thompson sampling variant for combinatorial allocation problems with complex capacity constraints—including refugee resettlement and social housing. They provide finite-sample regret bounds that are independent of the exponentially large number of possible allocations, demonstrating how bandit methods can scale to rich combinatorial structure in resource assignment problems.

https://doi.org/10.3390/bdcc10070206

Candelieri et al. [39] introduce a BO framework that maps the original optimization problem into the space of discrete probability distributions endowed with a Wasserstein metric. The GP surrogate and acquisition function are defined on this non-Euclidean manifold, and the acquisition minimizer is mapped back to the original space via a neural network. Computational results show that exploration in the Wasserstein space is significantly more effective than standard BO in the Euclidean space, with the advantage growing with problem dimensionality. Candelieri et al. [40], in another work, extend this framework by fully embedding the optimization process in the probability simplex equipped with the optimal transport distance. Both the GP surrogate and the acquisition function are generalized to functionals over the simplex, optimized via either automatic differentiation or a proximal-point gradient flow algorithm, yielding improved performance over standard BO that increases with problem dimensionality. While these works establish the Wasserstein simplex as a viable BO design space, they do not address the sequential resource allocation problem specifically, nor do they consider the SBF setting, time-varying budgets, or the multi-armed allocation structure that characterizes BORA's problem domain.

Xie and Chen [41] address BO in high-dimensional permutation spaces, introducing the Merge Kernel—derived from the divide-and-conquer structure of merge sort—as a scalable alternative to the Mallows kernel. The Merge Kernel achieves competitive performance in low dimensions and substantially outperforms Mallows in both optimization quality and computational efficiency as dimensionality grows, highlighting the importance of kernel design matched to the geometry of the search space. Oliveira et al. [42] propose Generative Bayesian Optimization (GBO), replacing the standard surrogate-plus-acquisition-function paradigm with generative models trained directly on utility values. Inspired by direct preference optimization, GBO scales naturally to large batches, non-continuous design spaces, and high-dimensional combinatorial settings, extending BO beyond the classical GP surrogate framework.

Chok and Vasil [43] propose the Cauchy-Simplex, an iteration scheme for convex optimization over the probability simplex. The simplex is mapped to the positive quadrant of the unit sphere, gradient descent is performed in latent space, and the result is projected back in a way that depends only on simplex variables. The method admits rigorous convergence guarantees expressed in terms of cross-entropy and KL divergence, and its per-iteration simplicity makes it well-suited for high-dimensional problems. Lanzetti et al. [44] develop a variational calculus framework in the Wasserstein space, combining ideas from optimal transport, variational analysis, and Wasserstein gradient flows. They derive first-order optimality conditions for optimization over probability measures that formally mirror Euclidean analogues such as the KKT conditions. The framework yields closed-form solutions in several cases and supports the design of computationally tractable algorithms, with applications to machine learning, drug discovery, and distributionally robust optimization. Floto et al. [45] propose performing diffusion on the probability simplex via the softmax of an Ornstein–Uhlenbeck process, yielding a generative model over categorical distributions through the logistic-normal distribution. The approach provides a continuous probabilistic framework over the simplex and is shown to extend naturally to diffusion on the unit cube, with applications to bounded image generation.

The key contributions of this work can be outlined as follows:

1. The original formulation of the optimal resource allocation problem [21,22] is extended to accommodate (i) a time-varying budget, denoted as $\bar{b}^t$, and (ii) an equality constraint expressed as $\sum_{i=1:m} x_i^t = \bar{b}^t$.
2. A new Bayesian Optimization framework for sequential optimal Resource Allocation—referred to as BORA—is introduced as a viable alternative to SBF, particularly in practical scenarios where SBF proves inapplicable. The novelty does not lie in

applying BO to resource allocation per se, but in the principled reformulation of the problem over the probability simplex equipped with the Wasserstein distance. Prior BO approaches, including CBO, operate in Euclidean spaces and cannot natively respect the simplex structure without ad hoc projections that discard geometric information. BORA is the first framework to define the GP surrogate and acquisition function directly and consistently within the Wasserstein geometry of the simplex. To navigate the newly defined design space, a suitable measure of distance between distributions is required. The Wasserstein distance, also known as optimal transport distance, is adopted owing to its favorable mathematical properties and the efficient computational methods it supports.

3. Both the Gaussian Process and the acquisition function are extended into an acquisition functional defined over the probability simplex. The acquisition function is optimized over the simplex via a constrained gradient-based strategy. The Wasserstein gradient flow and proximal-point scheme are presented as a theoretical framework for optimization in probability spaces and identified as a direction for future empirical investigation.
4. Embedding the Bayesian Optimization framework within the probability simplex yields an efficient algorithm whose performance gains over conventional Bayesian optimization become more pronounced as problem dimensionality grows.

3. Background

3.1. Optimal Resource Allocation via SBF (Brief Overview)

The core challenge of the optimal resource allocation problem is clearly explained in [21,46,47], and can be summarized as follows: allocating an excessive amount of resources to a job i ensures its completion, but yields no insight into its level of difficulty (i.e., the value of v_i). Conversely, assigning insufficient resources may prevent the job from being completed, yet it provides useful information about its difficulty while conserving resources that can be used to explore other arms.

The practical motivation for studying optimal resource allocation through SBF originates from the cache allocation problem. In this setting, jobs correspond to computer processes, and a job is considered completed at a given time step t if no cache misses occur. As noted in [21], similar formulations apply to other contexts, including load balancing in networked systems and various computing scenarios where limited resources—such as bandwidth, radio spectrum, or CPU capacity—must be distributed efficiently among processes. Along with the introduction of the problem [21], and successively [22], propose an optimistic SBF algorithm to solve the optimal resource allocation problem (introduced in Section 6), with a constant budget over time rescaled to 1. Their algorithm was inspired by the optimal policy, which can be applied in the case of known $v_1, \ldots, v_m$. This policy consists of fully allocating resources to the job with the smallest v_i and then allocating the remaining resources to the next easiest job. Starting from that, in the proposed SBF algorithm, at each time step t, the unknown v_i is replaced

with a high-probability lower bound $v_i^t \leq vi$. This corresponds to the optimistic strategy, which assumes that each job is as easy as reasonably possible. Naturally, learning a reliable confidence interval about every v_i is delicate. The immediate reward function $f(x^t) = \sum_{i=1}^{m} \mathcal{B}(x_i^t / v_i)$; it is non-differentiable at $x_i^t = v_i$, and for $x_i^t \geq v_i$ the job will always be completed, but providing little information about the job's difficulty. This issue is addressed by using the lower estimate of every v_i. Furthermore, since x_i^t varies over time, the observations of the successfully completed jobs are not identically distributed. This means that a standard sample average estimator is weak. On the contrary, the SBF algorithm estimates the lower and upper confidence bounds of each v_i, denoted with $\underline{v}_i$ and $\bar{v}_i$, which are, respectively, non-decreasing and non-increasing.

https://doi.org/10.3390/bdcc10070206

This is sufficient to guarantee that the SBF algorithm is numerically stable. The SBF algorithm and its improved version are reported, in pseudo-code, in [21,22], respectively.

3.2. Gaussian Process-Based Bayesian Optimization (Brief Overview)

A Gaussian Process (GP) can be interpreted as a set of random variables for which any finite subset follows a joint Gaussian distribution. It is fully specified by a mean function $\mu(x)$ and a covariance function $k(x,\acute{x})$, and is commonly denoted as $\mathcal{GP}(\mu(x), k(x,\acute{x}))$ [48,49]. By conditioning these functions on observed data, a GP can be employed as a probabilistic regression model to generate predictions at any input location x.

More specifically, assume that t observations $y^{1:t} = \{y^{(i)}\}_{i=1:t}$, have been collected by evaluating the function f(x) at input locations $X^{1:t} = \{X^{(i)}\}_{i=1:t}$. The notation $y^{(i)}$ is used to account for the possibility of noisy measurements, i.e., $y^{(i)} = f\,x^{(i)} + \varepsilon^{(i)}$, where $\varepsilon^{(i)}$ is assumed to be Gaussian noise with zero mean and variance σ_ε^2, such that $\varepsilon^{(i)} \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ for all $i \in \{1,\ldots,t\}$. Under these assumptions, the GP predictive mean and variance, conditioned on the observed data, are given by:

$$\mu(x) = k(x, X^{1:t})\,[K + \sigma_\varepsilon^2 I]^{-1} y^{(1:t)} \tag{5}$$

$$\sigma^2(x) = k(x,x) - k(x, X^{1:t})\,[K + \sigma_\varepsilon^2 I]^{-1}\,k(X^{1:t}, x) \tag{6}$$

Here, K denotes the covariance function of the GP, specified as a kernel function encoding structural assumptions about f, such as smoothness, and denotes a $t \times t$ matrix with entries $k_{i,j} = k\left(x^{(i)}, x^{(j)}\right)$, while $k(x, X^{1:t})$ represents a t-dimensional row vector $[k(x, x^{(1)}), \ldots, k(x, x^{(t)})]$, and T indicates the transpose. The predictive uncertainty of the GP is expressed as $\sigma(\boldsymbol{x}) = \sqrt{\sigma(x)^2}$.

Prior to conditioning on observations, two prior assumptions must be defined. Typically, the mean function is set to zero (without loss of generality, since the predictive mean in (5) is not restricted to this value), while the covariance function is selected from a class of kernel functions. These kernels encode structural assumptions about the function being modeled, particularly regarding smoothness [23,25,48,49]. In this work, the Squared Exponential (SE) kernel is adopted:

$$k(x,\acute{x}) = \sigma_f \cdot e^{-\frac{1}{2}\frac{\|x-\acute{x}\|^2}{\ell^2}} \tag{7}$$

where $\sigma_f \in \mathbb{R}$ and $\ell \in \mathbb{R}_+^m$ are hyperparameters controlling, respectively, the signal variance and the smoothness of the function. When all components satisfy $\ell_i = \bar{\ell}$, $\forall i \in \{1,\ldots,m\}$, the kernel is isotropic; otherwise, anisotropic behavior arises, allowing different smoothness levels across dimensions.

Since a GP provides both predictions and their associated uncertainty, it is naturally suited as a probabilistic regression framework. This characteristic is fundamental in addressing the exploration–exploitation trade-off in Bayesian Optimization (BO). At each iteration, an acquisition function (also referred to as a utility or infill function) is optimized to determine the next query point, balancing between exploiting current knowledge ("trusting the prediction") and exploring uncertain regions ("favoring uncertainty"). Formally, the following auxiliary optimization problem is solved:

$$x^{(t+1)} = arg \max_{x \in \Omega \subset \mathbb{R}^m} \mathcal{U}(x;\, \mu^{(t)}(x), \sigma^{(t)}(x)) \tag{8}$$

where $\mathcal{U}(x;\, \mu^{(t)}(x), \sigma^{(t)}(x))$ denotes the acquisition function, explicitly dependent on the current GP mean and uncertainty estimates. Various acquisition functions exist, offering different trade-offs between exploration and exploitation [23–25]. In this study, the well-known GP Upper Confidence Bound (GP-UCB) [28] is employed and will be described in detail later.

4. Wasserstein Distance and Optimal Transport

This section introduces the fundamental concepts underlying the Wasserstein (WST) distance, along with models used to compute it. It also discusses regularization strategies to address computational challenges, as well as methods for performing optimization within the Wasserstein space and ensuring the validity of Gaussian kernels in this setting.

4.1. Wasserstein Distance

The Wasserstein distance between two continuous probability distributions, denoted as $P^{(1)}$ and $P^{(2)}$, is defined as:

$$W_p(P^{(1)}, P^{(2)}) = \left(\inf_{\pi \in \Pi(P^{(1)}, P^{(2)})} \int_{\{\Gamma \times \Gamma\}} d\left(\gamma^{(1)}, \gamma^{(2)}\right)^p d\pi\left(\gamma^{(1)}, \gamma^{(2)}\right) \right)^{\frac{1}{p}} \quad (9)$$

Here, $d(\gamma^{(1)}, \gamma^{(2)})$ represents the ground metric, while $\Pi\left(P^{(1)}, P^{(2)}\right)$ denotes the set of all admissible joint distributions (transport plans) with marginals $P^{(1)}$ and $P^{(2)}$. The parameter $p > 1$ determines the order of the distance.

In the special case of one-dimensional distributions, the Wasserstein distance admits a closed-form expression. Let $P^{(1)}$ and $P^{(2)}$ be the cumulative distribution functions associated with distributions $\hat{P}^{(1)}$ and $\hat{P}^{(2)}$, and let their inverses be the corresponding quantile functions. Then:

$$W_p(P^{(1)}, P^{(2)}) = \left(\int_0^1 \left| \left(\hat{P}^{(1)}\right)^{-1}\left(\gamma^{(1)}\right) - \left(\hat{P}^{(2)}\right)^{-1}\left(\gamma^{(2)}\right) \right|^p d\gamma \right)^{\frac{1}{p}} \quad (10)$$

For discrete probability measures of the form:

$$\alpha = \sum_{i=1}^{\eta} a_i \delta_{\gamma_i}, \beta = \sum_{j=1}^{v} b_j \delta_{\omega_j} \quad (11)$$

the optimal transport (OT) problem, following the Kantorovich formulation [50], is represented via a coupling matrix $P \in \mathbb{R}_+^{\eta \times v}$, where each element P_{ij} specifies the amount of mass transported from the location γ_i to ω_j.

Given probability vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, the set of all feasible transport plans is:

$$U(a, b) = \{P \in \mathbb{R}_+^{\eta \times v} : P1_v = \boldsymbol{a} \wedge P^T 1_\eta = b\} \quad (12)$$

with:

$$P1_v = \left(\sum_j P_{ij} \right)_i \in \mathbb{R}^n \; and \; P^T 1_v = \left(\sum_i P_{ij} \right)_j \in \mathbb{R}^v \quad (13)$$

The set $U(a, b)$ is convex and bounded, defined by $\eta + v$ linear constraints. The discrete OT problem can then be expressed as:

$$\mathcal{W}_p(\alpha, \beta) = L_C(a, b) = \min_{P \in U(a,b)} \sum_{ij} C_{ij} P_{ij} \quad (14)$$

where C_{ij} denotes the cost of transporting one unit of mass from location i to location j.

For a comprehensive treatment of Wasserstein spaces and their properties (e.g., convergence and completeness), see [51–53]. Additional analysis on optimal transport over probability simplices defined on graphs is provided in [54].

It is worth noting that computing the Wasserstein distance is generally computationally intensive. However, certain cases—such as one-dimensional empirical distributions or Gaussian distributions—admit simpler solutions [55].

4.2. Entropic Regularization

Directly solving the OT problem in (14) can be computationally expensive. A common approach to mitigate this is to introduce an entropic regularization term based on the entropy of the coupling matrix P, (i.e., $H(P)$), to the term $W_p(\alpha, \beta)$, defined as:

$$H(P) = \sum_{ij} P_{ij} log\ P_{ij} \tag{15}$$

Incorporating this term leads to the regularized problem:

$$\begin{aligned} \mathcal{W}_p(\alpha, \beta) = & \min_{P \in \mathbb{R}_+^{n \times n'}} \sum_{i=1}^{n} \sum_{j=1}^{n'} \left(P_{ij}\, C_{ij} + \lambda P_{ij} \log P_{ij}\right) \\ s.t. \sum_{j=1}^{n'} P_{ij} = a_i, & \quad \sum_{i=1}^{n} P_{ij} = b_j, \quad P_{ij} \geq 0 \end{aligned} \tag{16}$$

This formulation, known as Sinkhorn regularization, provides an efficient approximation method [55]. The regularization parameter λ must be chosen carefully: large values may overly smooth the solution, while small values can lead to numerical instability and reduced scalability.

4.3. Wasserstein Gradient Flow and Proximal Algorithm

For a smooth function $F(\mathbf{x})$: $\mathbb{R}^d \to \mathbb{R}$, classical gradient descent takes the form:

$$\mathrm{x}^{(l+1)} = \mathrm{x}^{(l)} - \tau \nabla F(\mathrm{x}) \tag{17}$$

where τ is the step size and $\nabla F(\mathbf{x}) = \left(\frac{\partial F}{\partial x_1}, \ldots, \frac{\partial F}{\partial x_\eta}\right)$.

When dealing with non-smooth functions, the proximal-point method [56] can be used:

$$\mathrm{x}^{(l+1)} = \underset{\mathrm{x} \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{2} \left|\left|\mathrm{x} - \mathrm{x}^{(l)}\right|\right|^2 + \tau F(\mathrm{x}) \tag{18}$$

This idea extends naturally to optimization over probability simplices by replacing the Euclidean distance with the Wasserstein distance. For a function $F(\mathbf{a})$ defined over the simplex Σ_η, the update rule becomes:

$$\mathrm{a}^{(l+1)} = \underset{\mathrm{a} \in \Sigma_\eta}{\operatorname{argmin}} \frac{1}{2} \mathcal{W}_2\left(\mathrm{a}, \mathrm{a}^{(l)}\right)^2 + \tau F(\mathrm{a}) \tag{19}$$

This formulation can be further generalized to measures:

$$\alpha^{(l+1)} = \underset{\alpha \in \mathcal{M}_+^1(\Gamma)}{\operatorname{argmin}} \frac{1}{2} \mathcal{W}_2\left(\alpha, \alpha^{(l)}\right)^2 + \tau F(\alpha) \tag{20}$$

where $F(\alpha)$ is a functional over probability measures ($\mathcal{M}_+^1(\Gamma)$). This makes the proximal-point method particularly suitable for optimizing acquisition functions in probability spaces [57]. On the other hand, this schema can easily make the optimization of the acquisition function computationally expensive and, subsequently, limit its adoption to problems whose objective function is largely more expensive.

We will empirically prove that embedding the Wasserstein distance into the GP's kernel and optimizing the acquisition function as usual—but constrained to the probability simplex—allows us to obtain better results than standard BO without significantly affecting the computational cost.

4.4. Positive Definiteness of Kernels in the Wasserstein Space

Two problems arise in designing a BO framework upon Wasserstein spaces: the kernel of the Gaussian Process and the optimization of the acquisition function. In this section, we shall deal with the kernel.

When constructing a BO framework in Wasserstein spaces, two main challenges arise: defining a valid kernel for the Gaussian Process and optimizing the acquisition function.

Focusing on the kernel, a common choice is $k_w(w, w') = e^{-\frac{\mathcal{W}_2^2(w,w')}{2\ell^2}}$. However, unlike Euclidean spaces, Wasserstein spaces are not Hilbert spaces. As a result, kernels based on Wasserstein distances are not guaranteed to be positive definite in general. Ensuring positive definiteness requires additional conditions, which are only satisfied in specific cases. A sufficient condition ensuring positive definiteness occurs when the considered histograms are one-dimensional, which is the case for the optimization framework adopted in this work. In particular, within the proposed setting, the histograms remain univariate regardless of the dimensionality of the original search space X. An approximation result in [58] shows that, as the kernel length-scale parameter ℓ approaches zero, the minimum eigenvalue of the kernel matrix $K(W_{1:n}, W_{1:n})$ whose entries are defined as $k_{ij} = k(w_i, w_j)$, converges to 1, thereby guaranteeing that the kernel is positive definite.

A different perspective arises in the statistical literature, where the data are directly modeled as probability density functions. In this context, an isometric relationship is established between the Wasserstein space and a closed convex subset of the space of square-integrable functions [59].

Additionally, the geometry of the search space is explicitly taken into account when optimizing the acquisition function. The algorithm proposed in this direction, namely Geometry-aware Bayesian Optimization (GaBO), leads to conclusions consistent with those of Feragen, particularly regarding the evaluation of the probability that a kernel is positive definite as a function of its length-scale parameter. The central idea when operating on manifolds is that a Riemannian manifold locally behaves like a Euclidean space around each point.

In the specific case of univariate probability measures, and for $0 \leq p \leq 2$, the Wasserstein distance $\mathcal{W}_p(\alpha, \beta)$ possesses a Hilbertian structure. In particular, when $p = 2$, the Wasserstein-based Squared Exponential kernel is guaranteed to be positive definite and is defined as:

$$k_{WSE}(\alpha, \acute{\alpha}) = \sigma_f \cdot e^{-\frac{1}{2}\frac{W_p^2(\alpha,\acute{\alpha})}{\ell^2}} \tag{21}$$

where ℓ is the length-scale parameter and $\alpha, \beta \in M_+^1(\mathbb{R})$. This result implies that $k_{WSE}(.,.)$ can be safely adopted as a kernel function for Gaussian Processes defined over $M_+^1(\mathbb{R})$. It should be noted, however, that extending this construction to multivariate probability distributions is feasible but considerably more involved and does not admit a straightforward formulation.

5. BO Algorithms for Optimal Resource Allocation

This section introduces three different BO-based methods designed to address the optimal sequential resource allocation problem. The first approach is a variant of Constrained Bayesian Optimization (CBO), adapted to account for a budget constraint that varies over time. The other two methods—referred to as BORA$_{SE}$ and BORA$_{WSE}$—handle the same constraint by reformulating the problem within the probability simplex. The key distinction between these two variants lies in the kernel used within the GP: BORA$_{SE}$ relies on a standard SE kernel, whereas BORA$_{WSE}$ employs a Wasserstein-based SE kernel, which is more appropriate when dealing with probability distributions.

5.1. Constrained Bayesian Optimization

At iteration t, the algorithm has access to the dataset $D_{1:t} = \{X_{1:t}, y_{1:t}\}$, where $X_{1:t} = \{X^{(i)}\}_{i=1:t}$ are the evaluated points and $\boldsymbol{y}_{1:t} = \{y^{(i)}\}_{i=1:t}$ are the corresponding observations. Additionally, a budget $\bar{b}^{(t+1)}$ is available for the next allocation step.

The GP model is constructed using this dataset, with hyperparameters of the SE kernel—namely the signal variance $\sigma_f \in \mathbb{R}_+$ and the length-scale vector $\ell \in \mathbb{R}_+^m$—estimated via maximum likelihood. This yields predictive mean and variance functions $\mu^t(x)$ and $\sigma^t(x)$.

The next allocation $x^{(t+1)}$ is then obtained by solving the following constrained optimization problem:

$$x^{(t+1)} = arg \max_{x \in \Omega \subset \mathbb{R}^m} \mu^{(t)}(x) + \beta^{(t)}\sigma^{(t)}(x) \quad s.t. \sum_{i=1:m} x_i = \bar{b}^{(t+1)} \tag{22}$$

where β is a parameter regulating the balance between pure exploitation (i.e., $\beta = 0$) and pure random search (i.e., $\beta \to \infty$). Although [28] originally proposed a logarithmic scheduling—along with a convergence proof, under a limited number of queries—more recently [60] reported better performances by randomly sampling β from a given distribution, outperforming the original scheduling on a range of synthetic and real-world problems.

Although the GP is defined on the entire search space Ω, the next promising allocation decision, $x^{(t+1)}$, is selected according to step 2 of the algorithm. In a simple case, with just 2 arms, this means that the amount $x_1^{(t+1)} + x_2^{(t+1)}$ must match the available budget, leading UCB to be maximized over a specific line.

5.2. BORA$_{SE}$: BO over the Probability Simplex via SE Kernel

The information available to BORA$_{SE}$ coincides with that used in CBO. However, BORA$_{SE}$ introduces an additional dataset, denoted as $A_{1:t}$, which corresponds to a transformation of $X_{1:t}$ into elements of the probability simplex:

$$\Delta_{m-1} = \left\{ a \in \mathbb{R}_+^m, \sum_{j=1:m} a_j = 1 \right\} \tag{23}$$

More specifically, each element of $A_{1:t}$ is obtained through the following normalization:

$$a^{(i)} = x^{(i)}/\bar{b}^{(t)} \tag{24}$$

Thus, each vector $a^{(i)}$ can be interpreted as the set of weights of a corresponding univariate discrete probability measure $\alpha^{(i)}$, defined over the fixed support $\{1, \ldots, m\}$.

Transforming the original data points from the domain Ω into probability distributions within the simplex Δ_{m-1}, as described in (24), leads to several important observations:

- The simplex Δ_{m-1} forms a linear subspace of dimension $m - 1$, i.e., it has one dimension less than the original search space.
- Distinct points $x^{(i)}$ and $x^{(j)}$ in the original space may correspond to the same point a in the simplex, depending on their associated budgets $\bar{b}^{(i)}$ and $\bar{b}^{(j)}$. For instance, $x^{(i)} = (5; 3; 1)$ and $x^{(j)} = (10; 6; 2)$, with $\bar{b}^{(i)} = 9$ and $\bar{b}^{(j)} = 18$, both yield $a^{(i)} = a^{(j)} = (5/9; 1/3; 1/9)$. As a consequence, the observations $y_{1:t}$ must be treated as noisy, even in the case where the original function f(x) is noise-free.

The BORA$_{SE}$ method determines the next evaluation point $x^{(t+1)}$ through the following procedure:

1. A GP model is trained using the dataset $(A_{1:t},\ y_{1:t})$, where the hyperparameters of the SE kernel, namely $\sigma_f \in R_+$ and $\ell \in \mathbb{R}_+^m$, are estimated via maximum likelihood. The resulting GP is defined over the simplex Δ_{m-1}, meaning that in Equation (21), the variables $x, \acute{x} \in \Omega$ are replaced by $a, \acute{a} \in \Delta_{m-1}$.
2. The next candidate point is obtained by maximizing the GP-UCB acquisition function over the simplex:

$$a^{(t+1)} = arg \max_{a \in \Delta_{m-1}} \mu^{(t)}(a) + \sqrt{\beta^{(t)}}\sigma^{(t)}(a) \quad (25)$$

3. The optimal $a^{(t+1)}$ is then mapped back to the original domain through the available budget, $x^{(t+1)} = \bar{b}^{(t+1)}a^{(t+1)}$. It is important to emphasize that, unlike CBO, the BORA$_{SE}$ algorithm operates directly within the probability simplex. Consequently, the probability distributions $a^{(1)}, \cdots, a^{(t)}$ are converted back into points $x^{(1)}, \cdots, x^{(t)}$ using their corresponding budgets $\bar{b}^{(1)}, \cdots, \bar{b}^{(t)}$, as previously described.

5.3. BORA$_{WSE}$: BO over Probability Simplex via Wasserstein-SE Kernel

The third method, referred to as *BORA$_{WSE}$*, follows the same sequence of steps as *BORA$_{SE}$*. The primary difference lies in the choice of kernel used within the GP model. Since the optimization is performed over the simplex Δ_{m-1}, whose elements represent univariate discrete probability distributions, it is natural to adopt a distance measure specifically suited for such objects.

In recent years, the Wasserstein distance [61], together with Optimal Transport theory [55], has gained increasing attention due to its effectiveness in a wide range of applications, including imaging, signal processing, natural language processing, learning, and optimization [62–65].

In essence, the Wasserstein distance quantifies the discrepancy between two probability distributions α and $\acute{\alpha}$, regardless of whether their supports are discrete or continuous. The main idea is to transform one distribution into the other by transporting probability mass, with each movement incurring a cost determined by a ground metric on the support. The Wasserstein distance corresponds to the minimum total cost required to perform this transformation. It is typically denoted as $W_p(\alpha, \acute{\alpha})$, where $0 < p < \infty$ determines the order of the metric.

In this work, the focus is on univariate discrete probability measures. Such a measure α is defined by a fixed one-dimensional support $\{1, \ldots, m\}$ and an associated weight vector $a \in \Delta_{m-1}$. It is worth recalling that, in this framework, the weight vectors are obtained from the original query points via Equation (24).

Consistent with this setting, we rely on the result in [55]. While the Wasserstein distance is not Hilbertian in general, there exist particular cases where it admits a closed-form expression and becomes Hilbertian. Specifically, we consider:

(i) Univariate probability measures with fixed support $\{1, \ldots, m\}$.
(ii) A binary ground metric, where transporting one unit of mass from index i to j incurs a cost of 1 if $j \neq i$, and 0 otherwise.

Under these assumptions, the Wasserstein distance takes the following closed form:

$$W_p(\alpha, \alpha') = \left[\frac{1}{2}\sum_{i=1}^{m}|a_i - a_i'|\right]^{\frac{1}{p}} \quad (26)$$

and is therefore Hilbertian. This property is crucial, since Hilbertian distances can be used to construct radial basis function kernels. In particular, for any Hilbertian distance d, the function $e^{-d^q/\lambda}$ defines a positive definite kernel for $0 \leq q \leq 2$ and $\lambda > 0$.

Based on this result, the Wasserstein-based Squared Exponential kernel is defined in (21). In summary, the BORA$_{WSE}$ algorithm follows the same procedure as BORA$_{SE}$, with the key difference being the use of $k_{WSE}(\alpha, \alpha')$ instead of the standard SE kernel

$k_{SE}(x, x')$. This change is particularly important because it influences how the immediate reward is approximated.

It is important to recall that the immediate reward $\sum_{i=1}^{2} \mathcal{B} x_i^{(t)}/v_i$ is not differentiable at $x_i^{(t)} = v_i$. This non-smooth behavior is better captured by the GP model when using the Wasserstein-based kernel, as in BORA$_{WSE}$.

Finally, BORA$_{WSE}$ selects the next point $x^{(t+1)}$ as follows:

- A GP is trained using $(\mathcal{A}_{1:t}, y_{1:t})$, where the hyperparameters of the Wasserstein-SE kernel, $\lambda \in \mathbb{R}_+^m$, are estimated via maximum likelihood. The GP is defined over the simplex Δ_{m-1} using the Wasserstein distance.
- The GP-UCB acquisition function is maximized over the simplex:

$$a^{(t+1)} = arg \max_{a \in \Delta_{m-1}} \mu^{(t)}(a) + \sqrt{\beta^{(t)}} \sigma^{(t)}(a)$$

- The resulting $a^{(t+1)}$ is mapped back to the original space using the available budget:

$$x^{(t+1)} = \overline{b}^{(t+1)} a^{(t+1)}.$$

6. Experiments and Results

6.1. Case 1: Optimal Sequential Allocation of Computing Resources

This experimental scenario is motivated by prior studies presented in [21,22,59]. The problem consists of distributing computational resources $x^{(t)}$ among m jobs (with $i = 1, \ldots, m$) at each time step $t = 1, \ldots, T$. The objective is to maximize the cumulative number of successfully completed jobs over time. The immediate reward function is defined as:

$$f(x^{(t)}) = \sum_{i=1}^{m} \mathcal{B}(x_i^{(t)}/v_i) \quad (27)$$

where the parameters v_i are unknown to the decision-maker.

Compared to the original formulation, two key differences are introduced:

(i) The total available budget is allowed to vary over time, denoted by $\overline{b}^{(t)}$.
(ii) An equality constraint is imposed instead of an inequality constraint.

These modifications are motivated by practical considerations commonly encountered in real-world applications. The resulting optimization problem can therefore be expressed as:

$$\max_{x^{(t)} \in \mathbb{R}_+^m} \mathbb{E} \sum_{t=1}^{T} f(x^{(t)}) \quad s.t. \quad \sum_{i=1}^{m} x_i^{(t)} = \bar{b}^{(t)} \quad \forall t \quad (28)$$

Following the setup in [21], we initially focus on the case where $m = 2$. The experiments are divided into two distinct configurations:

- **Case 1.a:** The budget $\bar{b}^{(t)}$ is randomly selected initially and then remains constant across all time steps, i.e., $\bar{b}^{(t)} = \overline{b}$ for all $t = 1, \ldots, T$. Under this assumption, the values can be normalized such that $\overline{b} = 1$. Consequently, the only distinction from the original SBF framework lies in the use of an equality constraint.
- **Case 1.b:** The budget $\overline{b}(t)$ varies over time, with each value independently drawn from a uniform distribution over the interval [10, 100].

In both configurations, the time horizon is set to [10, 100], with parameters $\nu_1 = 25$ and $\nu_2 = 50$. To reduce the influence of stochastic variability, each experiment is repeated over 30 independent runs for every algorithm and test scenario.

First, we summarize the overall results related to the cumulative reward provided by the algorithms, over 30 independent runs, and for the two sub-settings considered. According to the results in Table 1, SBF is always outperformed by all of the three BO algorithms. On the other hand, BORA$_{WSE}$ slightly underperforms against CBO and BORA$_{SE}$ in Case1.a (i.e., under constant budget). However, it results in the best approach in Case1.b (i.e., budget changing over time), meaning that the GP using the Wasserstein SE kernel is better able to deal with the uncertainty implied by the random modifications of the budget over time.

Table 1. Final cumulative reward collected by the four algorithms over 30 independent runs. Results are expressed as "median [1st quantile; 3rd quantile]" (the higher the better).

Budget Type	m	CBO	BORA$_{SE}$	BORA$_{WSE}$	SBF
Constant	2	**2.05** [1.94; 2.19]	2.04 [1.94; 2.20]	1.93 [1.88; 2.00]	1.74 [1.56; 1.91]
Constant	10	**2.91** [2.87; 2.96]	**2.91** [2.86; 2.95]	2.85 [2.83; 2.88]	2.50 [2.42; 2.52]
Constant	15	**3.00** [2.92; 3.10]	2.99 [2.89; 3.09]	2.94 [2.90; 2.96]	2.69 [2.66; 2.70]
Constant	20	**3.07** [2.97; 3.19]	3.06 [2.94; 3.17]	2.98 [2.98; 3.04]	2.81 [2.78; 2.83]
Changing	2	**2.13** [2.11; 2.14]	2.10 [2.07; 2.11]	2.11 [2.10; 2.11]	1.95 [1.84; 2.04]
Changing	10	2.87 [2.86; 2.88]	2.90 [2.88; 2.91]	**2.91** [2.91; 2.91]	2.54 [2.48; 2.55]
Changing	15	3.00 [2.98; 3.02]	2.99 [2.97; 3.00]	**3.02** [3.02; 3.02]	2.63 [2.62; 2.67]
Changing	20	3.08 [3.08; 3.10]	3.07 [3.06; 3.08]	**3.11** [3.09; 3.11]	2.81 [2.76; 2.88]

After these general considerations, we specifically analyze the behaviors of the four algorithms by plotting the evolution of their cumulative rewards over time—as median and range 1st–3rd quantile—along with a boxplot of their final cumulative rewards, visually more explanatory than results in the previous table. For readability, charts and discussion are divided with respect to the two sub-settings

Results for Case 1.a (constant budget over time). Figures 1–4 refer to the setting with a constant budget over time for 2, 10, 15, and 20 arms, respectively. Overall, it is easy to understand that BO algorithms outperform SBF over the entire time window, and not only in terms of final cumulative reward. Moreover, as a general effect, increasing the number of arms leads to a reduction in the variability of performances, more relevant for SBF and BORA$_{WSE}$, especially in terms of final cumulative reward.

The 'utopian' curve is also reported, that is, the cumulative reward under complete knowledge; the closer to the utopian curve, the better. As already discussed, CBO resulted in the best option, even if the performances provided by BORA$_{SE}$ and BORA$_{WSE}$ are not so different.

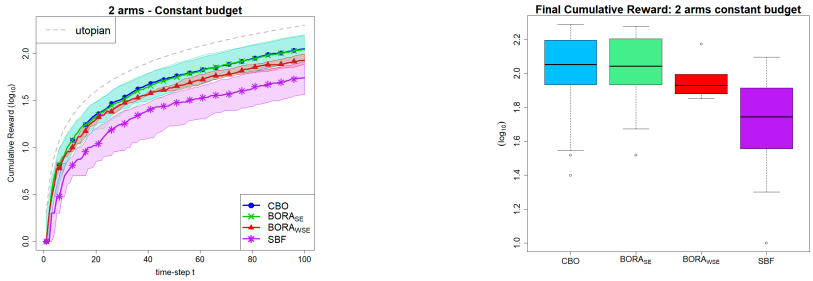

Figure 1. Case 1.a (i.e., constant budget over time) with number of arms m = 2. On the **left**, cumulative (log10) reward over time (median, 1st and 3rd quantile); on the **right**, box-plot of the final cumulative (log10) rewards of the four analyzed algorithms (the higher the better).

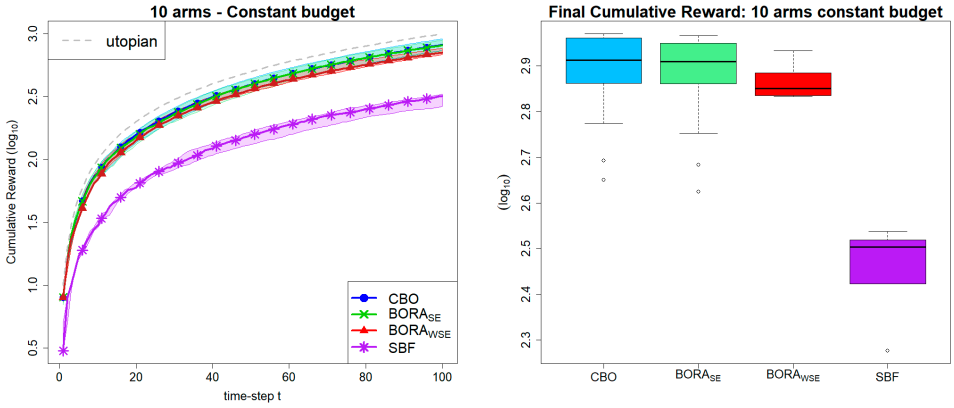

Figure 2. Case 1.a (i.e., constant budget over time) with number of arms m = 10. On the **left**, cumulative (log10) reward over time (median, 1st and 3rd quantile); on the **right**, box-plot of the final cumulative (log10) rewards of the four analyzed algorithms (the higher the better).

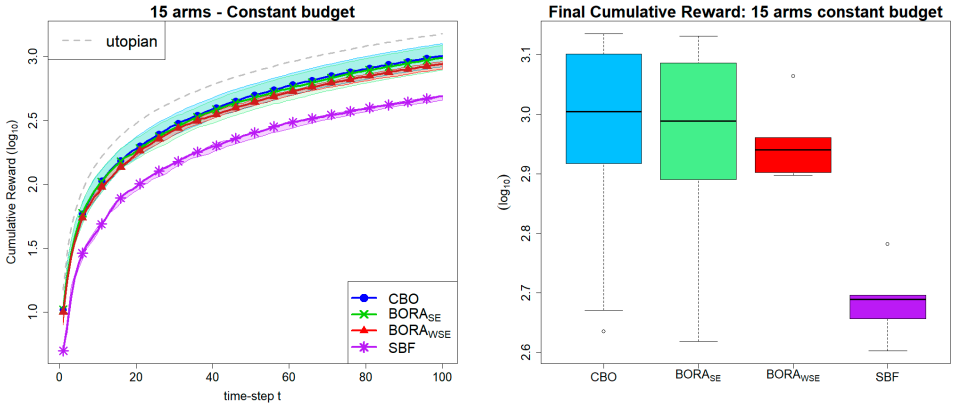

Figure 3. Case 1.a (i.e., constant budget over time) with number of arms m = 15. On the **left**, cumulative (log10) reward over time (median, 1st and 3rd quantile); on the **right**, box-plot of the final cumulative (log10) rewards of the four analyzed algorithms (the higher the better).

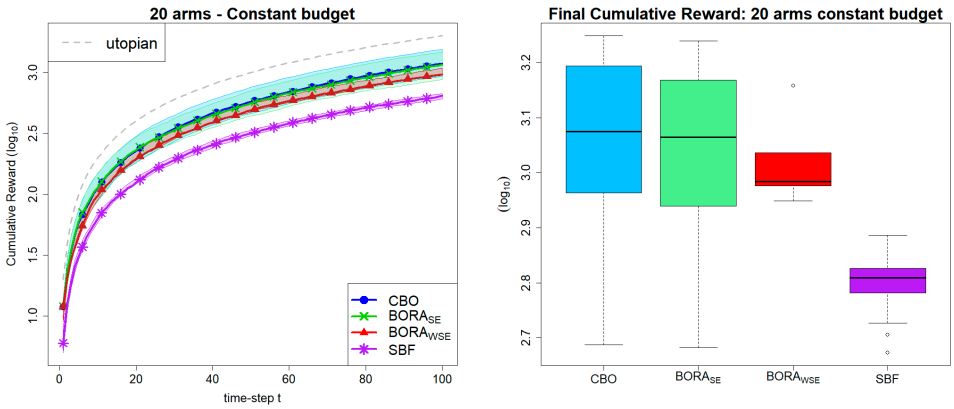

Figure 4. Case 1.a (i.e., constant budget over time) with number of arms m = 20. On the **left**, cumulative (log10) reward over time (median, 1st and 3rd quantile); on the **right**, box-plot of the final cumulative (log10) rewards of the four analyzed algorithms (the higher the better).

Results for Case 1.b (budget changing over time). Figures 5–8 refer to the setting with a constant budget over time, for 2, 10, 15, and 20 arms, respectively. Again, BO algorithms largely outperform SBF. The reduction in variability in the performances, with the number of arms increasing, can be observed again. The three BO algorithms continue to offer

similar performances, with BORA$_{WSE}$ gaining a slight but consistent advantage over CBO and BORA$_{SE}$ as the budget varies over time.

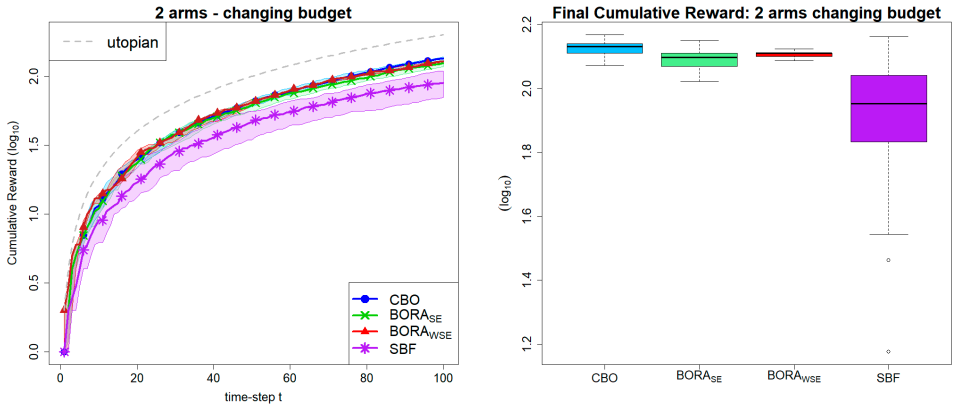

Figure 5. Case 1.b (i.e., budget changing over time) with number of arms m = 2. On the **left**, cumulative (log10) reward over time (median, 1st and 3rd quantile); on the **right**, box-plot of the final cumulative (log10) rewards of the four analyzed algorithms (the higher the better).

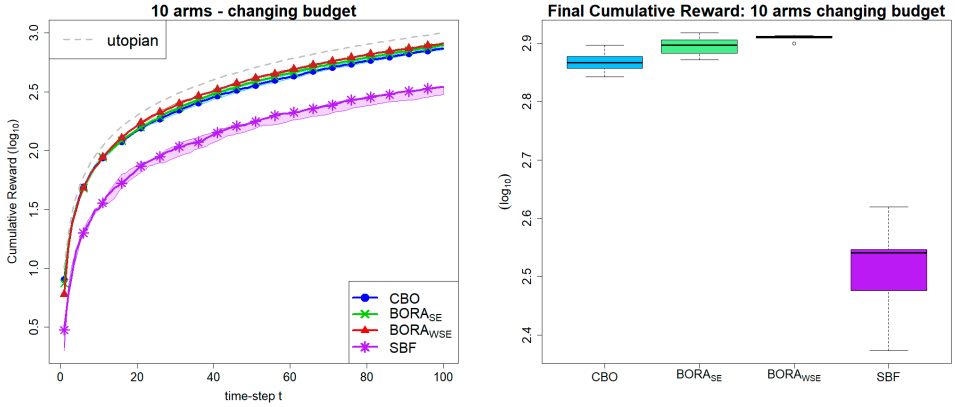

Figure 6. Case 1.b (i.e., budget changing over time) with number of arms m = 10. On the **left**, cumulative (log10) reward over time (median, 1st and 3rd quantile); on the **right**, box-plot of the final cumulative (log10) rewards of the four analyzed algorithms (the higher the better).

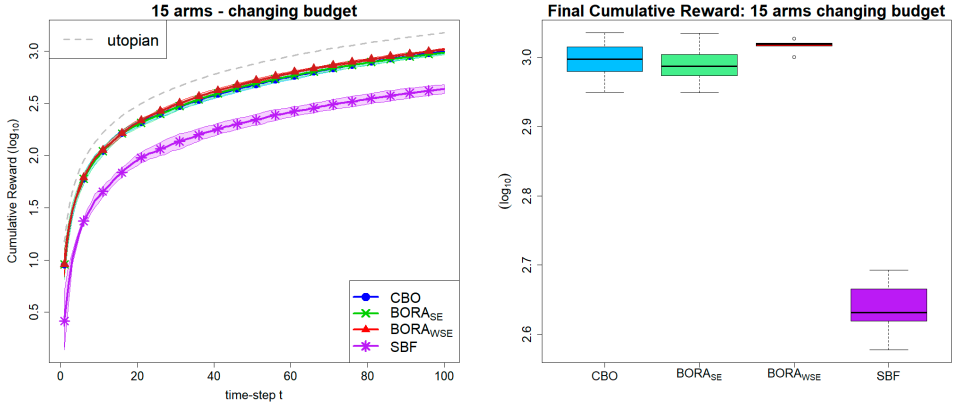

Figure 7. Case 1.b (i.e., budget changing over time) with number of arms m = 15. On the **left**, cumulative (log10) reward over time (median, 1st and 3rd quantile); on the **right**, box-plot of the final cumulative (log10) rewards of the four analyzed algorithms (the higher the better).

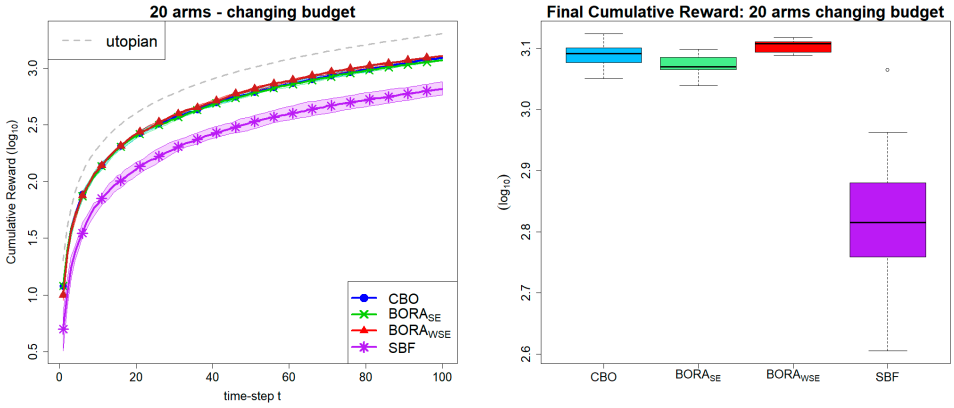

Figure 8. Case 1.b (i.e., budget changing over time) with number of arms m = 20. On the **left**, cumulative (log10) reward over time (median, 1st and 3rd quantile); on the **right**, box-plot of the final cumulative (log10) rewards of the four analyzed algorithms (the higher the better).

It is important to underline that, on average, BORAWSE shows an increase in cumulative reward when moving from Case 1.a to Case 1.b (i.e., its curve is closer to the utopian one under the same number of arms). This confirms that the Wasserstein-based kernel is better suited to handle the uncertainty introduced by time-varying budgets, giving BORAWSE a clear advantage over CBO and BORASE specifically in this setting.

6.2. Case 2: Multi-Channel Marketing

The second experimental scenario is motivated by a real-world problem involving the allocation of a marketing budget across multiple channels. The formulation remains consistent with (28); however, the immediate reward function is now expressed as:

$$f(x^{(t)}) = \sum_{i=1}^{m} \eta_i^{(t)} x_i^{(t)}$$

where $\eta_i^{(t)}$ denotes the (unknown) return generated by investing an amount $x_i^{(t)}$ in channel i at time step t. As in the previous case study, these parameters—here $\eta_i^{(t)}$—are not known to the learner. On the other hand, the available budget $\overline{b}^{(t)}$ is revealed before selecting the allocation $x^{(t)}$ at each time step.

The experimental setup considers a time horizon T = 100 and m = 15 distinct marketing channels. The budgets and returns are generated according to the following distributions:

$$\overline{b}^{(t)} \sim N(50, 10)$$

$$\eta_i^{(t)} max\{0, \sim N(\mu_{\eta_i}, \sigma_{\eta_i})\}$$

where the parameters μ_{η_i} and σ_{η_i} are themselves sampled as:

$$\mu_{\eta_i} \sim U(0, 1),\ \sigma_{\eta_i} \sim U(0, 0.2)$$

It is important to highlight several key differences compared to Case 1:

- The immediate reward, while still unknown, is now a general linear function, rather than a sum of binary outcomes (0–1 values).
- Due to this structural change, the SBF algorithm cannot be directly applied, as it relies on the assumption that each term in the reward function is binary.
- In contrast, both CBO and BORA can be employed without requiring any modification. These approaches are capable of modeling and optimizing the reward function

regardless of its specific form, since they only require access to the observed reward values rather than its individual components.

To reduce the impact of stochastic variability, 30 independent runs are conducted for each algorithm. As in the previous case study, two configurations are analyzed: one with a constant budget over time (Case 2.a) and another with a time-varying budget (Case 2.b).

Table 2 summarizes the overall results in terms of final cumulative rewards collected by the three BO algorithms, as median and 1st–3rd quantile over 30 independent runs. It is important to note that SBF cannot address this specific problem (at least in its original formulation). On this real-life application, BORA$_{WSE}$ resulted in the best algorithm, regardless of the number of arms.

Table 2. Final cumulative reward collected by the three BO algorithms over 30 independent runs. Results are expressed as "median [1st quantile; 3rd quantile]" (the higher the better). Remarkably, SBF cannot address this specific Case. Important: BORA$_{WSE}$ has been evaluated on only 5 independent runs because GP fitting and optimization of the acquisition function resulted particularly computationally expensive in this case study.

Budget Type	m	CBO	BORA$_{SE}$	BORA$_{WSE}$
Constant	5	3.37 [3.09; 3.51]	3.31 [3.04; 3.48]	**3.40** [3.22; 3.61]
Constant	10	3.27 [3.01; 3.47]	3.30 [3.13; 3.50]	**3.51** [2.77; 3.55]
Constant	15	3.44 [3.05; 3.60]	3.49 [3.15; 3.59]	**3.51** [3.33; 3.52]
Changing	5	3.33 [3.14; 3.53]	3.37 [3.13; 3.55]	**3.53** [3.22; 3.57]
Changing	10	3.28 [2.97; 3.49]	3.28 [2.97; 3.51]	**3.60** [3.32; 3.62]
Changing	15	3.37 [3.14; 3.60]	3.41 [3.14; 3.60]	**3.57** [3.45; 3.64]

As already done for the previous case study, now we analyze the behavior of the BO algorithms over time, separately for the sub-settings constant and changing budget.

Results for Case 2.a (constant budget over time). Figures 9–11 refer to the setting with a constant budget over time, for 5, 10, and 15 arms (i.e., reasonable numbers of marketing channels to consider in a real-life application), respectively. BORA$_{WSE}$ always outperforms CBO and BORA$_{SE}$. However, it is important to note that, contrary to the other two BO approaches, BORA$_{WSE}$ has been evaluated on only 5 independent runs because GP fitting and optimization of the acquisition function resulted particularly computationally expensive in this case study.

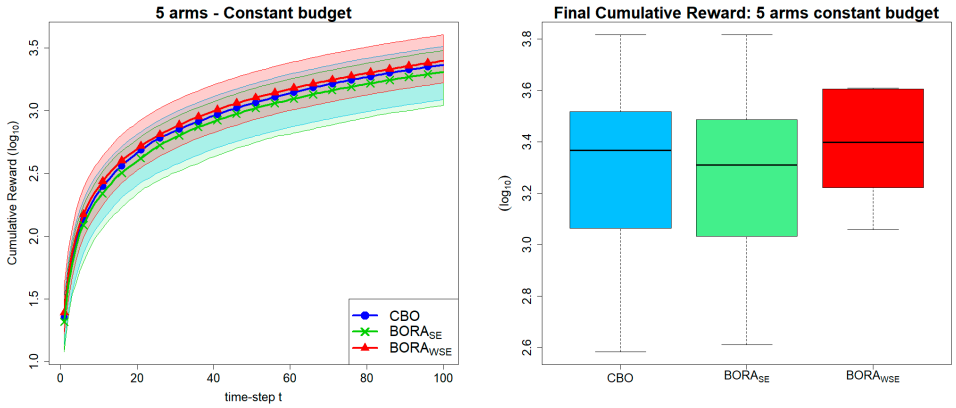

Figure 9. Case 2.a (i.e., constant budget over time) with number of arms m = 5. On the **left**, cumulative (log10) reward over time (median, 1st and 3rd quantile); on the **right**, box-plot of the final cumulative (log10) rewards of the four analyzed algorithms (the higher the better).

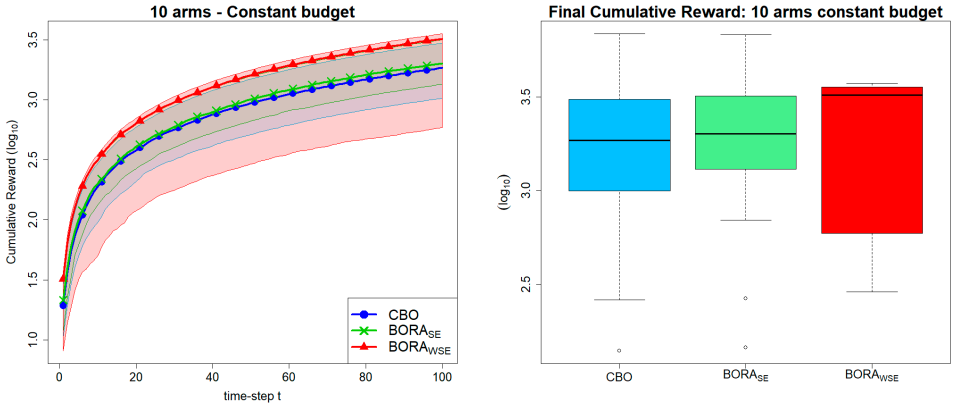

Figure 10. Case 2.a (i.e., constant budget over time) with number of arms m = 10. On the **left**, cumulative (log10) reward over time (median, 1st and 3rd quantile); on the **right**, box-plot of the final cumulative (log10) rewards of the four analyzed algorithms (the higher the better).

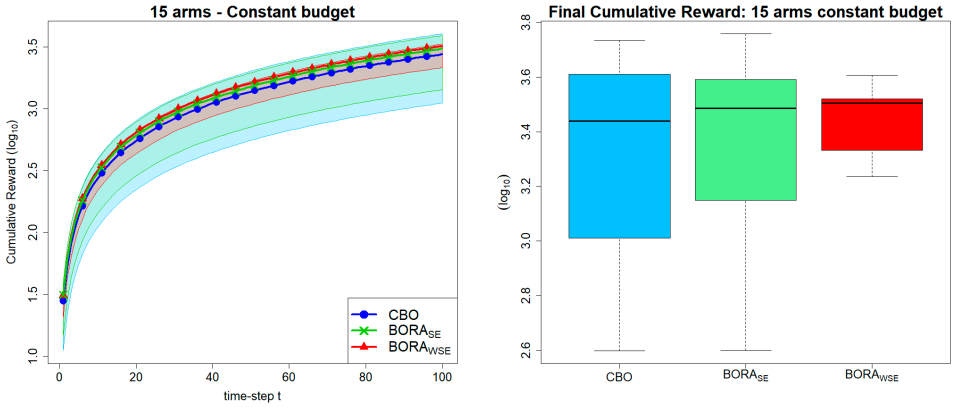

Figure 11. Case 2.a (i.e., constant budget over time) with number of arms m = 20. On the **left**, cumulative (log10) reward over time (median, 1st and 3rd quantile); on the **right**, box-plot of the final cumulative (log10) rewards of the four analyzed algorithms (the higher the better).

Results for Case 2.b (budget changing over time). Figures 12–14 refer to the setting with budget changing over time, for 5, 10, and 15 arms, respectively. Moving from constant to changing budget just confirms previous results, with BORAWSE always outperforming CBO and BORASE. However, this better performance entails an additional computational cost for BORAWSE, because the GP fitting procedure and the optimization of UCB become more expensive for this case study.

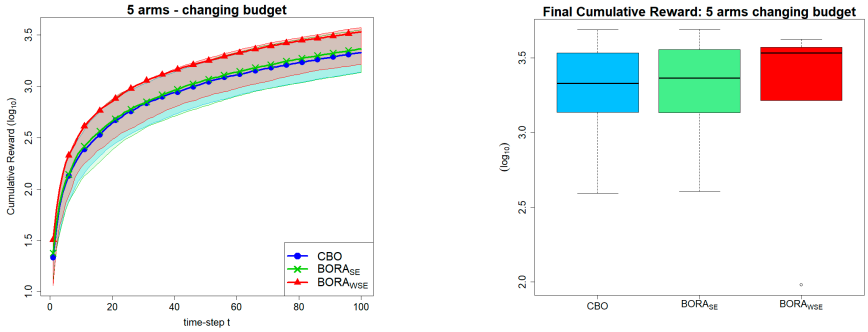

Figure 12. Case 2.b (i.e., budget changing over time) with number of arms m = 5. On the **left**, cumulative (log10) reward over time (median, 1st and 3rd quantile); on the **right**, box-plot of the final cumulative (log10) rewards of the four analyzed algorithms (the higher the better).

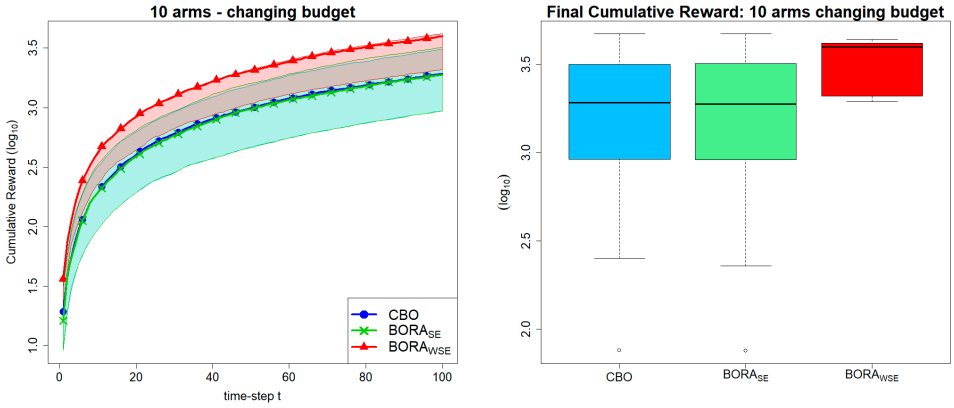

Figure 13. Case 2.b (i.e., budget changing over time) with number of arms m = 10. On the **left**, cumulative (log10) reward over time (median, 1st and 3rd quantile); on the **right**, box-plot of the final cumulative (log10) rewards of the four analyzed algorithms (the higher the better).

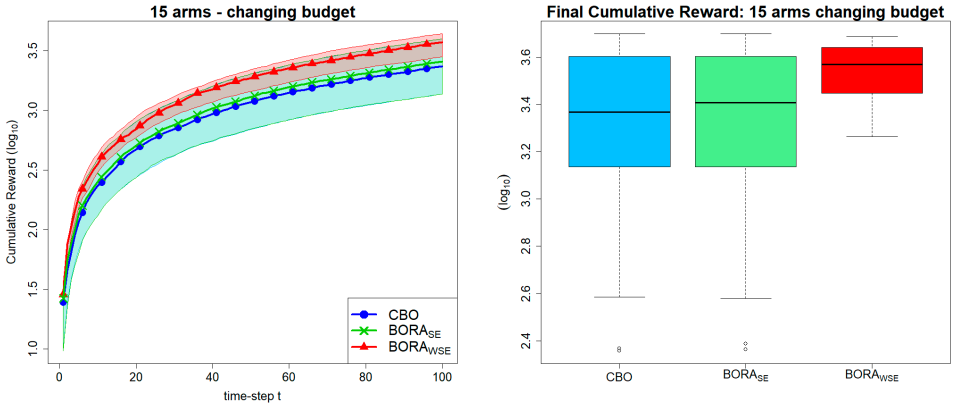

Figure 14. Case 2.b (i.e., budget changing over time) with number of arms m = 15. On the **left**, cumulative (log10) reward over time (median, 1st and 3rd quantile); on the **right**, box-plot of the final cumulative (log10) rewards of the four analyzed algorithms (the higher the better).

A final and important consideration concerns computational time. For a single run, on average, BORA$_{SE}$ requires a time lower than CBO (approximately one third). BORA$_{WSE}$ is the most computationally expensive: it could require minutes instead of seconds, compared against CBO and BORA, but it achieves competitive or superior performance, particularly under time-varying budget settings and in the multi-channel marketing application, where the Wasserstein-based kernel better captures the distributional structure of the problem. Finally, although SBF is the fastest algorithm, it cannot address the mixed marketing application, namely Case 2, and provides a significantly lower cumulative reward than all three BO algorithms.

7. Conclusions

This work investigates the optimal resource allocation problem initially introduced in [21], extending it to two practical application domains. The first scenario builds upon the original formulation used to motivate the SBF approach, while the second concerns multi-channel marketing. In addition, the study considers a time-varying budget, representing a more general and realistic setting compared to the original SBF formulation.

The results show that the three proposed Bayesian Optimization-based approaches—namely CBO, BORA$_{SE}$, and BORA$_{WSE}$—achieve better performance than SBF. Furthermore, it is demonstrated that certain application domains, such as multi-channel

marketing, cannot be effectively handled using SBF alone. This highlights that CBO and BORA provide more flexible and powerful frameworks for combined learning and optimization tasks.

Another key finding is that BORA offers a substantial computational advantage over CBO. This improvement is mainly due to the fact that BORA operates within the probability simplex and leverages a Wasserstein-based Squared Exponential kernel.

A potential drawback of both CBO and BORA is that they may not be suitable for scenarios where the time interval between consecutive decision steps is extremely short (e.g., less than one second). However, for the types of applications considered in this study, where sufficient time is available between steps t and t + 1, these methods can be effectively employed in place of SBF. It should also be noted that BORA$_{WSE}$ was evaluated on only five independent runs due to its high computational cost, compared to 30 runs for CBO and BORA$_{SE}$. The comparison between BORA$_{WSE}$ and the other algorithms is therefore based on samples of unequal size, and a fully uniform experimental protocol would be desirable in future work.

From a theoretical standpoint, deriving regret bounds for BORA remains an open problem. While GP-UCB achieves sublinear regret in Euclidean spaces [28], extending such guarantees to the probability simplex equipped with the Wasserstein distance is nontrivial, due to the non-stationarity of the Wasserstein SE kernel and the non-Hilbertian geometry of the search space [57]. Characterizing the maximum information gain for the Wasserstein SE kernel, which is the key quantity controlling regret in the framework of [28], constitutes an important direction for future theoretical work.

Several promising directions emerge from the results presented in this paper. A first extension concerns replacing the GP surrogate with generative models trained directly on observed utility values, as proposed by Oliveira et al. [42], thereby enabling large-batch optimization over non-continuous and high-dimensional combinatorial spaces. A second direction involves the automatic design of BO kernels via Large Language Models, as explored by Suwandi et al. [66], which could yield more expressive and problem-tailored surrogates. Complementarily, Tang and Paulson [67] propose integrating LLMs with evolutionary computation through crossover and mutation operators in the algorithmic space, opening the way to cost-aware, multi-fidelity BO strategies, particularly relevant when computational resources are limited. Diffusion-enabled optimal transport distances for graph matching represent a further promising direction, where diffusion dynamics enrich the Wasserstein geometry to capture relational structure, potentially extending the framework to allocation problems defined over graphs. Finally, Beatty [68] extends optimal transport distances to quantum structures, suggesting a longer-term research avenue for resource allocation in quantum computing settings, with connections to functional inequalities, many-body physics, and quantum generative models.

Author Contributions: Conceptualization, F.A. and A.C.; methodology, F.A. and A.C.; software, A.C.; validation, A.C., I.S. and A.P.; writing—original draft preparation, A.C., F.A. and A.P.; writing—review and editing, A.C., F.A. and I.S.; visualization, A.C., I.S. and A.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The entire code, including the acquisition functions, is freely available at the following repository: https://github.com/acandelieri/BORA_BDCC_journal.git (accessed on 25 March 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Gunjan, A.; Bhattacharyya, S. A brief review of portfolio optimization techniques. *Artif. Intell. Rev.* **2023**, *56*, 3847–3886. https://doi.org/10.1007/s10462-022-10273-7.
2. Xidonas, P.; Steuer, R.; Hassapis, C. Robust portfolio optimization: A categorized bibliographic review. *Ann. Oper. Res.* **2020**, *292*, 533–552. https://doi.org/10.1007/s10479-020-03630-8.
3. Zhang, S.; Huang, K.; Yuan, Y. Spare parts inventory management: A literature review. *Sustainability* **2021**, *13*, 2460. https://doi.org/10.3390/su13052460.
4. Ziukov, S. A literature review on models of inventory management under uncertainty. *Verslo Sist. Ekon.* **2015**, *5*, 26–35. https://doi.org/10.13165/VSE-15-5-1-03.
5. Bakker, H.; Dunke, F.; Nickel, S. A structuring review on multi-stage optimization under uncertainty: Aligning concepts from theory and practice. *Omega* **2020**, *96*, 102080. https://doi.org/10.1016/j.omega.2019.06.006.
6. Carpentier, P.; Chancelier, J.-P.; Cohen, G.; De Lara, M. *Stochastic Multi-Stage Optimization: At the Crossroads Between Discrete Time Stochastic Control and Stochastic Programming*; Probability Theory and Stochastic Modelling; Springer International Publishing: Cham, Switzerland, 2015; Volume 75. https://doi.org/10.1007/978-3-319-18138-7.
7. Seyedi, I.; Hamedi, M.; Tavakkoli-Moghaddam, R. Optimization for a truck scheduling problem in multi-door cross dokcing with learning effect and deteriorating jobs. *J. Transp. Res.* **2022**, *19*, 183–206. Available online: https://www.trijournal.ir/article_119317_en.html (accessed on 18 February 2026)
8. Lattimore, T.; Szepesvári, C. *Bandit Algorithms*; Cambridge University Press: Cambridge, UK, 2020.
9. Slivkins, A. Introduction to multi-armed bandits. *Found. Trends® Mach. Learn.* **2019**, *12*, 1–286. https://doi.org/10.1561/2200000068.
10. Balcan, M.-F.; Dick, T.; Pegden, W. Semi-bandit optimization in the dispersed setting. In *Conference on Uncertainty in Artificial Intelligence*; PMLR: Cambridge, MA, USA, 2020; pp. 909–918.
11. Chen, W.; Wang, L.; Zhao, H.; Zheng, K. Combinatorial semi-bandit in the non-stationary environment. In *Uncertainty in Artificial Intelligence*; PMLR: Cambridge, MA, USA, 2021; pp. 865–875. Available online: https://proceedings.mlr.press/v161/chen21a.html (accessed on 18 February 2026).
12. Jourdan, M.; Mutnỳ, M.; Kirschner, J.; Krause, A. Efficient pure exploration for combinatorial bandits with semi-bandit feedback. In *Algorithmic Learning Theory*; PMLR: Cambridge, MA, USA, 2021; pp. 805–849.
13. Neu, G.; Bartók, G. An Efficient Algorithm for Learning with Semi-bandit Feedback. In *Algorithmic Learning Theory*; Jain, S., Munos, R., Stephan, F., Zeugmann, T., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2013; Volume 8139, pp. 234–248. https://doi.org/10.1007/978-3-642-40935-6_17.
14. Wen, Z.; Kveton, B.; Ashkan, A. Efficient learning in large-scale combinatorial semi-bandits. In *International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2015; pp. 1113–1122. Available online: https://proceedings.mlr.press/v37/wen15.html (accessed on 18 February 2026).
15. Sonkar, S.K.; Kharat, M.U. A review on resource allocation and VM scheduling techniques and a model for efficient resource management in cloud computing environment. In *2016 International Conference on ICT in Business Industry & Government (ICTBIG)*; IEEE: New York, NY, USA, 2016; pp. 1–7.
16. Thananjeyan, B.; Kandasamy, K.; Stoica, I.; Jordan, M.; Goldberg, K.; Gonzalez, J. Resource allocation in multi-armed bandit exploration: Overcoming sublinear scaling with adaptive parallelism. In *International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2021; pp. 10236–10246. Available online: http://proceedings.mlr.press/v139/thananjeyan21a.html (accessed on 18 February 2026).
17. Vinothina, V.V.; Sridaran, R.; Ganapathi, P. A survey on resource allocation strategies in cloud computing. *Int. J. Adv. Comput. Sci. Appl.* **2012**, *3*, 97–104.
18. Wang, S.; Chen, W. Thompson sampling for combinatorial semi-bandits. In *International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2018; pp. 5114–5122. Available online: https://proceedings.mlr.press/v80/wang18a.html (accessed on 18 February 2026).
19. Yousafzai, A.; Gani, A.; Noor, R.M.; Sookhak, M.; Talebian, H.; Shiraz, M.; Khan, M.K. Cloud resource allocation schemes: Review, taxonomy, and opportunities. *Knowl. Inf. Syst.* **2017**, *50*, 347–381. https://doi.org/10.1007/s10115-016-0951-y.
20. Barrier, A.; Garivier, A.; Stoltz, G. On best-arm identification with a fixed budget in non-parametric multi-armed bandits. In *International Conference on Algorithmic Learning Theory*; PMLR: Cambridge, MA, USA, 2023; pp. 136–181.
21. Lattimore, T.; Crammer, K.; Szepesvári, C. Optimal Resource Allocation with Semi-Bandit Feedback. *arXiv* **2014**, arXiv:1406.3840. https://doi.org/10.48550/arXiv.1406.3840.

22. Dagan, Y.; Koby, C. A better resource allocation algorithm with semi-bandit feedback. In *Algorithmic Learning Theory*; PMLR: Cambridge, MA, USA, 2018; pp. 268–320.
23. Archetti, F.; Candelieri, A. *Bayesian Optimization and Data Science*; SpringerBriefs in Optimization; Springer International Publishing: Cham, Switzerland, 2019. https://doi.org/10.1007/978-3-030-24494-1.
24. Candelieri, A. A gentle introduction to bayesian optimization. In *2021 Winter Simulation Conference (WSC)*; IEEE: New York, NY, USA, 2021; pp. 1–16.
25. Frazier, P.I. Bayesian Optimization. In *Recent Advances in Optimization and Modeling of Contemporary Problems*; Gel, E., Ntaimo, L., Shier, D., Greenberg, H.J., Eds.; INFORMS: Catonsville, MD, USA, 2018; pp. 255–278. https://doi.org/10.1287/educ.2018.0188.
26. Seyedi, I.; Candelieri, A.; Archetti, F. Distributionally Robust Bayesian Optimization via Sinkhorn-based Wasserstein Barycenter. *Mach. Learn. Knowl. Extr.* **2025**, *7*, 90. https://doi.org/10.3390/make7030090.
27. Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R.P.; De Freitas, N. Taking the human out of the loop: A review of Bayesian optimization. *Proc. IEEE* **2015**, *104*, 148–175.
28. Srinivas, N.; Krause, A.; Kakade, S.M.; Seeger, M.W. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Trans. Inf. Theory* **2012**, *58*, 3250–3265.
29. Auer, P.; Cesa-Bianchi, N.; Fischer, P. Finite-time Analysis of the Multiarmed Bandit Problem. *Mach. Learn.* **2002**, *47*, 235–256. https://doi.org/10.1023/A:1013689704352.
30. Gelbart, M.A.; Snoek, J.; Adams, R.P. Bayesian Optimization with Unknown Constraints. *arXiv* **2014**, arXiv:1403.5607. https://doi.org/10.48550/arXiv.1403.5607.
31. Gelbart, M.A. Constrained Bayesian Optimization and Applications. Ph.D. Thesis, Harvard University, Cambridge, MA, USA, 2015. Available online: https://lips.cs.princeton.edu/pdfs/gelbart2015thesis.pdf (accessed on 18 February 2026).
32. Letham, B.; Karrer, B.; Ottoni, G.; Bakshy, E. Constrained Bayesian Optimization with Noisy Experiments. *Bayesian Anal.* **2019**, *14*, 495–519. https://doi.org/10.1214/18-BA1110.
33. Candelieri, A. Resource Allocation via Bayesian Optimization: An Efficient Alternative to Semi-Bandit Feedback. In *Numerical Computations: Theory and Algorithms*; Sergeyev, Y.D., Kvasov, D.E., Astorino, A., Eds.; Springer Nature: Cham, Switzerland, 2025; pp. 34–48. https://doi.org/10.1007/978-3-031-81241-5_3.
34. Patriksson, M. A survey on the continuous nonlinear resource allocation problem. *Eur. J. Oper. Res.* **2008**, *185*, 1–46.
35. Brandt, J.; Bengs, V.; Haddenhorst, B.; Hüllermeier, E. Finding optimal arms in non-stochastic combinatorial bandits with semi-bandit feedback and finite budget. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2022; Volume 35, pp. 20621–20634.
36. Verma, A.; Hanawal, M.; Rajkumar, A.; Sankaran, R. Censored semi-bandits: A framework for resource allocation with censored feedback. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32. Available online: https://proceedings.neurips.cc/paper/2019/hash/3e91970f771a2c473ae36b60d1146068-Abstract.html (accessed on 18 February 2026).
37. Zuo, J.; Joe-Wong, C. Combinatorial multi-armed bandits for resource allocation. In *2021 55th Annual Conference on Information Sciences and Systems (CISS)*; IEEE: New York, NY, USA, 2021; pp. 1–4. Available online: https://ieeexplore.ieee.org/abstract/document/9400228/ (accessed on 26 February 2026).
38. Kasy, M.; Teytelboym, A. Matching with semi-bandits. *Econom. J.* **2023**, *26*, 45–66.
39. Candelieri, A.; Ponti, A.; Archetti, F. Wasserstein enabled Bayesian optimization of composite functions. *J. Ambient. Intell. Humaniz. Comput.* **2023**, *14*, 11263–11271. https://doi.org/10.1007/s12652-023-04640-7.
40. Candelieri, A.; Ponti, A.; Archetti, F. Bayesian optimization over the probability simplex. *Ann. Math. Artif. Intell.* **2025**, *93*, 77–91. https://doi.org/10.1007/s10472-023-09883-w.
41. Xie, Z.; Chen, L. From Sorting Algorithms to Scalable Kernels: Bayesian Optimization in High-Dimensional Permutation Spaces. *arXiv* **2025**, arXiv:2507.13263. https://doi.org/10.48550/arXiv.2507.13263.
42. Oliveira, R.; Steinberg, D.M.; Bonilla, E.V. Generative Bayesian Optimization: Generative Models as Acquisition Functions. *arXiv* **2025**, arXiv:2510.25240. https://doi.org/10.48550/arXiv.2510.25240.
43. Chok, J.; Vasil, G.M. Optimization over a probability simplex. *J. Mach. Learn. Res.* **2025**, *26*, 1–35.
44. Lanzetti, N.; Terpin, A.; Dörfler, F. Variational Analysis in the Wasserstein Space. *arXiv* **2024**, arXiv:2406.10676. https://doi.org/10.48550/arXiv.2406.10676.
45. Floto, G.; Jonsson, T.; Nica, M.; Sanner, S.; Zhu, E.Z. Diffusion on the Probability Simplex. *arXiv* **2023**, arXiv:2309.02530. https://doi.org/10.48550/arXiv.2309.02530.

46. Seyedi, I.; Candelieri, A.; Messina, E.; Archetti, F. Wasserstein Distributionally Robust Optimization for Chance Constrained Facility Location Under Uncertain Demand. *Mathematics* **2025**, *13*, 2144. https://doi.org/10.3390/math13132144.
47. Seyedi, I.; Candelieri, A.; Archetti, F. Fused Unbalanced Gromov–Wasserstein-Based Network Distributional Resilience Analysis for Critical Infrastructure Assessment. *Mathematics* **2026**, *14*, 417. https://doi.org/10.3390/math14030417.
48. Gramacy, R.B. *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2020.
49. Williams, C.K.; Rasmussen, C.E. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006; Volume 2.
50. Kantorovich, L. On the transfer of masses (in Russian). *Dokl. Akad. Nauk.* **1942**, *37*, 227. Available online: https://cir.nii.ac.jp/crid/1370565168575910170 (accessed on 17 July 2025).
51. Villani, C. *Optimal Transport*; Grundlehren der Mathematischen Wissenschaften; Springer: Berlin/Heidelberg, Germany, 2009; Volume 338. https://doi.org/10.1007/978-3-540-71050-9.
52. Santambrogio, F. {Euclidean, metric, and Wasserstein} gradient flows: An overview. *Bull. Math. Sci.* **2017**, *7*, 87–154. https://doi.org/10.1007/s13373-017-0101-1.
53. Panaretos, V.M.; Zemel, Y. *An Invitation to Statistics in Wasserstein Space*; SpringerBriefs in Probability and Mathematical Statistics; Springer International Publishing: Cham, Switzerland, 2020. https://doi.org/10.1007/978-3-030-38438-8.
54. Li, W. Transport information geometry: Riemannian calculus on probability simplex. *Inf. Geom.* **2022**, *5*, 161–207. https://doi.org/10.1007/s41884-021-00059-1.
55. Peyré, G.; Cuturi, M. Computational optimal transport with applications to data sciences. *Found. Trends® Mach. Learn.* **2019**, *11*, 355–607. https://doi.org/10.1561/2200000073.
56. Rockafellar, R.T. Monotone Operators and the Proximal Point Algorithm. *SIAM J. Control Optim.* **1976**, *14*, 877–898. https://doi.org/10.1137/0314056.
57. Candelieri, A.; Ponti, A.; Archetti, F. Gaussian Process regression over discrete probability measures: On the non-stationarity relation between Euclidean and Wasserstein Squared Exponential Kernels. *J. Glob. Optim.* **2025**, *92*, 253–278. https://doi.org/10.1007/s10898-025-01463-y.
58. De Plaen, H.; Fanuel, M.; Suykens, J.A. Wasserstein exponential kernels. In *2020 International Joint Conference on Neural Networks (IJCNN)*; IEEE: New York, NY, USA, 2020; pp. 1–6. Available online: https://ieeexplore.ieee.org/abstract/document/9207630/ (accessed on 27 February 2026).
59. Bigot, J. Statistical data analysis in the Wasserstein space. *ESAIM Proc. Surv.* **2020**, *68*, 1–19.
60. Berk, J.; Gupta, S.; Rana, S.; Venkatesh, S. Randomised Gaussian Process Upper Confidence Bound for Bayesian Optimisation. *arXiv* **2020**, arXiv:2006.04296. https://doi.org/10.48550/arXiv.2006.04296.
61. Villani, C. The Wasserstein distances. In *Optimal Transport*; Grundlehren der mathematischen Wissenschaften; Springer: Berlin/Heidelberg, Germany, 2009; Volume 338, pp. 93–111. https://doi.org/10.1007/978-3-540-71050-9_6.
62. Liu, B.; Rao, Y.; Lu, J.; Zhou, J.; Hsieh, C.-J. Multi-proxy wasserstein classifier for image classification. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 8618–8626. Available online: https://ojs.aaai.org/index.php/AAAI/article/view/17045 (accessed on 18 February 2026).
63. Wang, C.; Long, S.; Zeng, R.; Lu, Y. Imputation Method for Fetal Heart Rate Signal Evaluation Based on Optimal Transport Theory. *SN Comput. Sci.* **2021**, *2*, 454. https://doi.org/10.1007/s42979-021-00805-3.
64. Zhang, D.; Yuan, Z.; Liu, Y.; Liu, H.; Zhuang, F.; Xiong, H.; Chen, H. Domain-oriented Language Modeling with Adaptive Hybrid Masking and Optimal Transport Alignment. In *27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*; ACM: New York, NY, USA, 2021; pp. 2145–2153. https://doi.org/10.1145/3447548.3467215.
65. Zhang, J.; Liu, T.; Tao, D. An optimal transport analysis on generalization in deep learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *34*, 2842–2853.
66. Suwandi, R.C.; Yin, F.; Wang, J.; Li, R.; Chang, T.-H.; Theodoridis, S. Adaptive Kernel Design for Bayesian Optimization Is a Piece of CAKE with LLMs. *arXiv* **2025**, arXiv:2509.17998. https://doi.org/10.48550/arXiv.2509.17998.

67. Tang, W.-T.; Paulson, J.A. CAGES: Cost-aware gradient entropy search for efficient local multi-fidelity Bayesian optimization. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*; IEEE: New York, NY, USA, 2024; pp. 1547–1552. Available online: Available: https://ieeexplore.ieee.org/abstract/document/10886516/ (accessed on 26 February 2026).
68. Beatty, E. Wasserstein Distances on Quantum Structures: An Overview. *arXiv* **2025**, arXiv:2506.09794. https://doi.org/10.48550/arXiv.2506.09794.