

Bayesian Generalized Network Design

Yuval Emek

Faculty of Industrial Engineering and Management, Technion, Haifa, Israel
yemek@technion.ac.il

Shay Kutten

Faculty of Industrial Engineering and Management, Technion, Haifa, Israel
kutten@ie.technion.ac.il

Ron Lavi

Faculty of Industrial Engineering and Management, Technion, Haifa, Israel
ronlavi@ie.technion.ac.il

Yangguang Shi

Faculty of Industrial Engineering and Management, Technion, Haifa, Israel
shiyangguang@campus.technion.ac.il

Abstract

We study network coordination problems, as captured by the setting of *generalized network design* (Emek et al., STOC 2018), in the face of uncertainty resulting from partial information that the network users hold regarding the actions of their peers. This uncertainty is formalized using Alon et al.'s *Bayesian ignorance* framework (TCS 2012). While the approach of Alon et al. is purely combinatorial, the current paper takes into account computational considerations: Our main technical contribution is the development of (strongly) polynomial time algorithms for local decision making in the face of Bayesian uncertainty.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis; Theory of computation → Mathematical optimization; Theory of computation → Algorithm design techniques

Keywords and phrases approximation algorithms, Bayesian competitive ratio, Bayesian ignorance, generalized network design, diseconomies of scale, energy consumption, smoothness, best response dynamics

Digital Object Identifier 10.4230/LIPIcs.ESA.2019.45

Related Version A full version of the paper is available at <https://arxiv.org/abs/1907.00484>.

Funding Yuval Emek: The work of Yuval Emek was supported in part by an Israeli Science Foundation grant number 1016/17.

Shay Kutten: The work of Shay Kutten was supported in part by a grant from the ministry of science in the program that is joint with JSPS and in part by the BSF.

Ron Lavi: The work of Ron Lavi was partially supported by the ISF-NSFC joint research program (grant No. 2560/17).

Yangguang Shi: The work of Yangguang Shi was partially supported at the Technion by a fellowship of the Israel Council for Higher Education.

1 Introduction

In real-life situations, network users are often required to coordinate actions for performance optimization. This challenging coordination task becomes even harder in the face of *uncertainty*, as users often act with partial information regarding their peers. Can users overcome their *local* views and reach a good *global* outcome? How far would this outcome be from optimal?



© Yuval Emek, Shay Kutten, Ron Lavi, and Yangguang Shi;
licensed under Creative Commons License CC-BY

27th Annual European Symposium on Algorithms (ESA 2019).

Editors: Michael A. Bender, Ola Svensson, and Grzegorz Herman; Article No. 45; pp. 45:1–45:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

For a formal treatment of the aforementioned questions, we adopt the *Bayesian ignorance* framework of Alon et al. [4]. Consider N agents in a *routing* scenario, where each agent $i \in [N]$ should decide on a (u_i, v_i) -path a_i in the network with the objective of minimizing some global *cost function* that depends on the links' load. The (u_i, v_i) pair, also referred to as the *type* of agent i , is drawn from a distribution p_i . All agents know this distribution, but the actual realization (u_i, v_i) of each agent i is only known to i herself.

Our goal is to construct a *strategy* for each agent i that determines her action a_i based only on her individual type (u_i, v_i) . These strategies are computed in a “preprocessing stage” and the actual decision making happens in real-time without further communication. We measure the quality of a tuple of strategies in terms of its *Bayesian competitive ratio (BCR)* defined as the ratio of the expected cost obtained by these strategies to that of an optimal solution computed by an omnipotent algorithm (refer to Sec. 1.1.1 for the exact definition). To the best of our knowledge, this algorithmic evaluation measure has not been studied so far.

Our main technical contribution is a generic framework that yields strongly polynomial-time algorithms constructing agent strategies with low BCR for *Bayesian generalized network design (BGND)* problems – a setting that includes routing and many other network coordination problems. Our framework assumes cost functions that exhibit *diseconomy of scale (DoS)* [5, 6, 28], capturing the power consumption of devices that employ the popular *speed scaling* technique.

1.1 Model

For clarity of the exposition, we start with the special case of Bayesian routing in Sec. 1.1.1 and then present the more general BGND setting in Sec. 1.1.2. Conceptually, the new algorithmic problem of Bayesian routing that we define here is related to oblivious routing [21, 17, 35], where routing requests should be performed without any knowledge about actual network traffic. This means that the routing path chosen for a routing request may only depend on the network structure and the other parameters of the problem. Oblivious algorithms are attractive as they can be implemented very efficiently in a distributed environment as they base routing decisions only on local knowledge. As will become formally clear below, Bayesian routing has a similar flavor, but with an important additional ingredient. We will assume that the algorithm is equipped with statistical (“Bayesian”) knowledge about network traffic. Thus, in a sense, we replace internal randomization techniques, that oblivious routing usually employs, with actual data, while still being oblivious to other actual routing decisions and thus still maintaining the locality principle.¹

1.1.1 Special Case: Bayesian Routing

In the *full information* variant of the *routing* problem, we are given a (directed or undirected) graph $G = (V, E)$ and a set of N agents, where each agent $i \in [N]$ is associated with a node pair $(u_i, v_i) \in V \times V$, referred to as the (routing) *request* of agent i . This request should be satisfied by choosing some (u_i, v_i) -path in G , referred to as the (feasible) *action* of agent i , and the collection of all such paths is denoted by A_i .

¹ This is different from stochastic network design as these algorithms are not oblivious. More details are given below.

Let $A = A_1 \times \cdots \times A_N$ be the collection of all *action profiles*. The *load* on edge $e \in E$ with respect to action profile $a \in A$, denoted by l_e^a , is defined to be the number of agents whose actions include e , that is, $l_e^a = |\{i \in [N] : e \in a_i\}|$. The cost incurred by load l_e^a on edge e is determined by an (edge specific) *superadditive* cost function $F_e : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}_{\geq 0}$ such that for any $l \geq 0$,

$$F_e(l) = \xi_e \cdot l^\alpha, \quad (1)$$

where $\xi_e > 0$ (a.k.a. the *speed scaling factor*) is a parameter of edge e and $\alpha > 1$ (a.k.a. the *load exponent*) is a global constant parameter. Such a superadditive cost function captures, for example, the power consumption of network devices employing the popular *speed scaling* technique [37, 25, 7, 29, 12, 3] that allows the device to adapt its power level to its actual load. In particular, for those network devices that employ the speed scaling technique, the value of α generally satisfies $1 < \alpha \leq 3$ [24, 36]. Another application of the cost function (1) with $\alpha = 2$ is to model the queuing delay of users in a TCP/IP communication networks [18]. The goal in the (full information) routing problem is to construct an action profile $a \in A$ with the objective of minimizing the total cost $C(a) = \sum_{e \in E} F_e(l_e^a)$.

1.1.1.1 Extending to Partial Information

In the current paper, we extend the full information routing problem to the *Bayesian routing* problem, where the request of agent $i \in [N]$ is not fully known to all other agents. In this problem variant, agent $i \in [N]$ is associated with a set T_i of *types* so that each type $t_i \in T_i$ specifies its own routing request $(u_i^{t_i}, v_i^{t_i}) \in V \times V$. Let $A_i^{t_i}$ be the set of all (feasible) actions for (the request of) type t_i , namely, all $(u_i^{t_i}, v_i^{t_i})$ -paths in G and let $A_i = \bigcup_{t_i \in T_i} A_i^{t_i}$.

Agent i is also associated with a *prior distribution* p_i over the types in T_i and the crux of the Bayesian routing problem is that agent i should decide on her action while knowing the realization of her own prior distribution p_i (that is, the routing request she should satisfy) but without knowing the realizations of the prior distributions of the other agents $j \neq i$. Formally, let $T = T_1 \times \cdots \times T_N$ be the collection of *type profiles* and $A = A_1 \times \cdots \times A_N$ be the collection of *action profiles*. The set of (feasible) action profiles for a type profile $t \in T$ is denoted by $A^t = A_1^{t_1} \times \cdots \times A_N^{t_N}$ and the prior distribution over the type profiles in T is denoted by p . In this paper, p is assumed to be a product distribution, i.e., the probability of type profile $t \in T$ is $p(t) = \prod_{i=1}^N p_i(t_i)$.

The goal in the Bayesian routing problem is to construct for each agent $i \in [N]$, a *strategy* $s_i : T_i \mapsto A_i$ that maps agent i 's realized type $t_i \in T_i$ to an action $a_i \in A_i^{t_i}$. We emphasize that the decision of agent i is taken irrespective of the other agents' realized types which are not (fully) known to agent i . Intuitively, a strategy s_i can be viewed as a *lookup table* constructed in the "preprocessing stage", and queried at real-time to determine a (fixed) path for every (u_i, v_i) pair associated with i (cf. *oblivious routing* [30, 35]).

The set of strategies available for agent i is denoted by S_i and $S = S_1 \times \cdots \times S_N$ denotes the set of *strategy profiles*. For each type profile $t \in T$, the strategy profile $s \in S$ determines an action profile $a = s(t) \in A$ defined so that $a_i = s_i(t_i)$, $i \in [N]$. Using this notation, the objective in the Bayesian routing problem is to construct a strategy profile $s \in S$ that minimizes the total cost $C(s) = \mathbb{E}_{t \sim p} [\sum_{e \in E} F_e(l_e^{s(t)})]$.

1.1.1.2 Bayesian Competitive Ratio

Consider an algorithm \mathcal{A} that given a Bayesian routing instance, constructs a strategy profile s . To evaluate the performance of \mathcal{A} , we compare the total cost $C(s)$ to $\mathbb{E}_{t \sim p} [\text{OPT}(t)]$, where $\text{OPT}(t) = \min_{a \in A^t} \sum_{e \in E} F_e(l_e^a)$ is the cost of an optimal action profile for the type profile $t \in T$. This can be regarded as the expectation, over the same prior distribution p , of the

total cost incurred by an omnipotent algorithm that has a global view of the whole type profile t and enjoys unlimited computational resources. The *Bayesian competitive ratio* (BCR) of algorithm \mathcal{A} is the smallest $\beta \geq 1$ such that for every Bayesian routing instance, the strategy profile s constructed by \mathcal{A} satisfies $C(s) \leq \beta \cdot \mathbb{E}_{t \sim p}[\text{OPT}(t)]$.

Alon et al. [4] introduced the related criterion of *Bayesian ignorance* defined as $\frac{C(s^*)}{\mathbb{E}_{t \sim p}[\text{OPT}(t)]}$, where $s^* = \text{argmin}_{s \in S} C(s)$ is an optimal strategy profile for the given instance. This criterion quantifies the implication of the agents' partial knowledge regarding the global system configuration, irrespective of the computational complexity of constructing this optimal strategy profile. By definition, for any strategy profile $s \in S$, $C(s) = \mathbb{E}_{t \sim p}[\sum_{e \in E} F_e(l_e^{s(t)})] \geq \mathbb{E}_{t \sim p}[\min_{a \in A^t} \sum_{e \in E} F_e(l_e^a)]$, which implies that the Bayesian ignorance is at least 1. Notice that the BCR is equivalent to the product of the *approximation ratio* $\frac{C(s)}{C(s^*)}$ and the Bayesian ignorance, therefore it evaluates the loss caused by both algorithmic (computational complexity) considerations and the absence of the global information. The first contribution of the current paper is cast in the following theorem.

► **Theorem 1.** *For the Bayesian routing problem, there exists an algorithm whose BCR depends only on the load exponent parameter α . This algorithm is fully combinatorial and runs in strongly polynomial time.*

We emphasize that the BCR of the algorithm promised in Theorem 1 is independent of the number of agents N , the underlying graph G , the speed scaling factors ξ_e , $e \in E$, and the probability distribution p . Therefore, as α is assumed to be a constant, so is the BCR.

1.1.2 Bayesian Generalized Network Design

1.1.2.1 Generalized Network Design

The (full information) routing problem has recently been generalized by Emek et al. [15] to the wider family of *generalized network design (GND)* problems. In its full information form (the form considered in [15]), a GND instance is defined over N agents and a set E of *resources*. Each agent $i \in [N]$ is associated with an abstract (not necessarily routing) request characterized by a set $A_i \subseteq 2^E$ of (feasible) actions out of which, some action $a_i \in A_i$ should be selected. As in the routing case, the action profile $a = (a_1, \dots, a_N)$ induces a load of $l_e^a = |\{i \in [N] : e \in a_i\}|$ on each resource $e \in E$ that subsequently incurs a cost of $F_e(l_e^a)$, where $F_e : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}_{\geq 0}$ is a resource specific cost function. The goal is to construct an action profile $a \in A = A_1 \times \dots \times A_N$ with the objective of minimizing the total cost $C(a) = \sum_{e \in E} F_e(l_e^a)$.

The request of agent $i \in [N]$ is said to be *succinctly represented* [15] if its corresponding action set A_i can be encoded using $\text{poly}(|E|)$ bits. Identifying the resource set E with the edge set of an underlying graph G , the routing requests defined in Sec. 1.1.1 are clearly succinctly represented since each A_i corresponds to the set of (u_i, v_i) -paths in G , hence A_i can be encoded by specifying u_i and v_i (and G). Other examples for succinctly represented requests, where the resource set E is identified with the edge set of an underlying (directed or undirected) graph $G = (V, E)$, include:

- *multi-routing* requests in directed or undirected graphs, where given a collection $D_i \subseteq V \times V$ of *terminal* pairs, the action set A_i consists of all edge subsets $F \subseteq E$ such that the subgraph (V, F) admits a (u, v) -path for every $(u, v) \in D_i$; and
- *set connectivity* (resp., *set strong connectivity*) in undirected (resp., directed) graphs, where given a set $T_i \subseteq V$ of *terminals*, the action set A_i consists of all edge subsets $F \subseteq E$ that induce on G a connected (resp., strongly connected) subgraph that spans T_i .

All requests mentioned (implicitly or explicitly) hereafter are assumed to be succinctly represented.

1.1.2.2 Bayesian GND

In the current paper, we extend the (full information) GND setting to *Bayesian GND* (BGND). This extension is analogous to the extension of full information routing to Bayesian routing as defined in Sec. 1.1.1. In particular, agent $i \in [N]$ is now associated with a set T_i of types, where each type $t_i \in T_i$ corresponds to a request whose action set is denoted by $A_i^{t_i}$, and a prior distribution p_i over the types in T_i . A strategy s_i of agent i is a function that maps the agent's realized type $t_i \in T_i$ to an action $s_i(t_i) \in A_i^{t_i}$.

Similarly to the notation introduced in Sec. 1.1.1, let $T = T_1 \times \cdots \times T_N$ be the set of type profiles. Let $A_i = \bigcup_{t_i \in T_i} A_i^{t_i}$ and let $A = A_1 \times \cdots \times A_N$ be the set of action profiles. Let S_i be the set of strategies available for agent i and let $S = S_1 \times \cdots \times S_N$ be the set of strategy profiles. Given a strategy profile $s \in S$ and a type profile $t \in T$, let $a = s(t) \in A$ be the action profile defined so that $a_i = s_i(t_i)$, $i \in [N]$. The goal in the BGND problem is to construct a strategy profile $s \in S$ with the objective of minimizing the total cost

$$C(s) = \mathbb{E}_{t \sim p} \left[\sum_{e \in E} F_e \left(l_e^{s(t)} \right) \right]. \quad (2)$$

The BCR of Algorithm \mathcal{A} is the smallest $\beta \geq 1$ such that for every BGND instance, the strategy profile $s \in S$ constructed by \mathcal{A} satisfies $C(s) \leq \beta \cdot \mathbb{E}_{t \sim p}[\text{OPT}(t)]$, where $\text{OPT}(t) = \min_{a \in A^t} \sum_{e \in E} F_e(l_e^a)$.

1.1.2.3 Generalized Cost Functions

In addition to the generalization of (full information) routing to GND, [15] also generalizes the cost functions defined in Eq. (1) to cost functions of the form

$$F_e(l) = \sum_{j \in [q]} \xi_{e,j} \cdot l^{\alpha_j}, \quad (3)$$

where q is a positive integer, $\xi_{e,j}$ is a positive real for every $e \in E$ and $j \in [q]$, and α_j is a constant real no smaller than 1 for every $j \in [q]$.² We define $\alpha_{\max} = \max_{j \in [q]} \alpha_j$ and assume hereafter that $\alpha_{\max} > 1$. As discussed in [15], this generalization of Eq. (3) is not only interesting from a theoretical perspective, but also makes the model more applicable to practical network energy saving applications. Indeed, in realistic communication networks, a link often consists of several different devices (e.g., transmitter/receiver, amplifier, adapter), all of which are operating when the link is in use. As their energy consumption can vary in terms of the load exponents and speed scaling factors [36], Eq. (3) may often provide a more accurate abstraction of the actual link's power consumption.

1.1.2.4 Action Oracles

For a BGND problem \mathcal{P} , this paper develops a framework which generates an algorithm with BCR $O(\varrho^{\alpha_{\max}})$ when provided with an *action ϱ -oracle* for \mathcal{P} . An action ϱ -oracle with parameter $\varrho \geq 1$ for BGND problem \mathcal{P} (cf. the *reply ϱ -oracles* of [15]) is a procedure that given agent $i \in [N]$, type $t_i \in T_i$, and a *weight* vector $w \in \mathbb{R}_{>0}^E$, generates an action $a_i \in A_i^{t_i}$ such that $\sum_{e \in a_i} w(e) \leq \varrho \cdot \sum_{e \in a'_i} w(e)$ for any action $a'_i \in A_i^{t_i}$. An *exact action oracle* is an action ϱ -oracle with parameter $\varrho = 1$.

² The cost functions considered in [15] have a fixed additional term, capturing the resource's *startup cost*, that makes them even more general. Due to technical difficulties, in the current paper we were not able to cope with this additional term.

Notice that the optimization problem behind the action oracle is *not* a BGND problem: It deals with a *single* type of a *single* agent and the role of the resource cost functions is now taken by the weight vector. These differences often make it possible to implement the action oracle with known (approximation) algorithms.

For example, the Bayesian routing problem, which requires paths between the given node pairs, admit an exact action oracle implemented using, e.g., Dijkstra's shortest path algorithm [14, 19]. In contrast, the BGND problem with set connectivity requests in undirected graphs (P1), the BGND problem with set strong connectivity requests in directed graphs (P2), the BGND problem with multi-routing requests in undirected graphs (P3), and the BGND problem with multi-routing requests in directed graphs (P4) do not admit exact action oracles unless $P = NP$ as these would imply exact (efficient) algorithms for the *Steiner tree*, *strongly connected Steiner subgraph*, *Steiner forest*, and *directed Steiner forest* problems, respectively. However, employing known approximation algorithms for the latter (Steiner) problems, one concludes that BGND problem (P1) admits an action ϱ -oracle for $\varrho \leq 1.39$ [9]; BGND problem (P2) admits an action ν^ϵ -oracle, where ν is the number of terminals [10]; BGND problem (P3) admits an action 2-oracle [1]; and BGND problem (P4) admits an action $k^{1/2+\epsilon}$ -oracle, where k is the number of terminal pairs [11]. This means, in particular, that BGND problems (P1) and (P3) always admit an action ϱ -oracle with a constant approximation ratio ϱ , whereas BGND problems (P2) and (P4) admit such an oracle when ν and k are fixed [1, 10, 11, 9]. The guarantees of our approximation framework are cast in the following theorem.

► **Theorem 2.** *Consider a BGND problem \mathcal{P} with an action ϱ -oracle $\mathcal{O}_{\mathcal{P}}$. When provided access to $\mathcal{O}_{\mathcal{P}}$, the framework proposed in this paper generates an algorithm $\mathcal{A}_{\mathcal{P}}$ whose BCR depends only on the load exponent parameters $\alpha_1, \dots, \alpha_q$ of Eq. (3). This framework is fully combinatorial and runs in strongly polynomial time, hence if $\mathcal{O}_{\mathcal{P}}$ can be implemented to run in strongly polynomial time, then so can $\mathcal{A}_{\mathcal{P}}$.*

Again, we emphasize that the BCR of the algorithm promised in Theorem 2 is independent of the number of agents N , the number of resources $|E|$, the speed scaling factors $\xi_{e,j}$, $j \in [q]$, $e \in E$, and the probability distribution p . Therefore, as $\alpha_1, \dots, \alpha_q$ are assumed to be constants, so is the BCR. Since the Bayesian routing problem admits an exact action oracle, Theorem 1 follows trivially from Theorem 2. Throughout the remainder of this paper, we focus on the BGND framework promised in Theorem 2.

1.2 Related Works

The technical framework that we use is inspired by [15]. Sec. 3 gives a detailed technical overview including a full comparison.

In the full information case, network design problems with superadditive cost functions as defined in Eq. (1) have been extensively studied with the motivation of improving the energy efficiency of networks [5, 6, 28]. To the best of our knowledge, none of these studies has been extended to the Bayesian case.

In the research works on oblivious routing (e.g., [17, 35, 27, 23]), the absence of global information in routing is modeled in an adversarial (non-Bayesian) manner. In particular, oblivious routing assumes that no knowledge about t_{-i} is available when determining every a_i , and the performance of the algorithm is evaluated by means of its *competitive ratio* $\max_{t \in T} \frac{\sum_{e \in E} F_e(l_e^s(t))}{\text{OPT}(t)}$. For the cost function $F_e(l) = l^\alpha$ with $\alpha > 1$, Englert and Räcke [17]

propose an $O(\log^\alpha |V|)$ -competitive oblivious routing algorithm for the scenario where the traffic requests are allowed to be partitioned into fractional flows. Shi et al. [35] prove that for such a cost function, there exists no oblivious routing algorithm with competitive ratio $O\left(|E|^{\frac{\alpha-1}{\alpha+1}}\right)$ when it is required to choose an integral path for every request.

The Bayesian approach is often used in the game theoretic literature to model the uncertainty a player experiences regarding the actions taken by the other players. Roughgarden [32] studies a *routing game* (among other things) in which the players share (equally) the cost of the edges they use and proposes a theoretical tool called *smoothness* to analyze the *price of anarchy* (PoA) of this game in a Bayesian setting, defined as $\frac{\max_{s \in S^{\text{BNE}}} C(s)}{\mathbb{E}_{t \sim p}[\text{OPT}(t)]}$, where S^{BNE} denotes the set of *Bayes-Nash equilibria*. In particular, he proves that with the cost function $F_e(l) = \xi_{e,1} \cdot l + \xi_{e,2} \cdot l^2$, the PoA is bounded by $\frac{5}{2}$. We employ the smoothness toolbox in our algorithmic construction, as further described in Sec. 5 (see also the overview in Sec. 3, as well as the detail in Sec. 6 of the full version [16]).

Alon et al. [4] investigate the Bayesian routing game with a constant cost function $F_e = \xi_e$ and prove that the Bayesian ignorance $\frac{C(s^*)}{\mathbb{E}_{t \sim p}[\text{OPT}(t)]}$ is bounded by $O(N)$ (resp., $O(\log |E|)$) in directed (resp., undirected) graphs $G = (V, E)$. They also introduce game theoretic variants of the Bayesian ignorance notion and analyze them in that game.

To deal with the inherent uncertainty of the demand in realistic networks, many research works have been conducted on *stochastic network design* [22, 13, 31], formulated as a *two-stage stochastic optimization* problem: in the first stage, each link in the network has a fixed cost and the algorithm needs to make decisions to purchase links knowing the probability distribution over the network demands; in the second stage, the network demands are realized (according to the aforementioned probability distribution) and should be satisfied, which may require purchasing additional links, this time with an inflated cost. The objective is to minimize the total cost of the two stages plus a load dependent term, in expectation.

The BGND setting considered in the current paper is different from two-stage stochastic optimization (particularly, stochastic network design) in several aspects, the most significant one is that in BGND, an agent’s strategy should dictate her “complete action” (e.g., a path for routing requests) for every possible type, obliviously of the realized types of the other agents. In particular, one cannot “update” the agents’ actions and purchase additional resources at a later stage to satisfy the realized demands. Moreover, the current paper evaluates the performance of a BGND algorithm by means of its BCR that takes into consideration computational complexity limitations as well as the lack of global information (see Sec. 1.1) whereas the literature on two-stage stochastic optimization typically evaluates algorithms using standard approximation guarantees that accounts only for computational complexity limitations.

In [20], Garg et al. investigate online combinatorial optimization problems where the requests arriving online are drawn independently and identically from a known distribution. As an example, Garg et al. [20] study the online Steiner tree problem on an undirected graph $G = (V, E)$. In this problem, at each step the algorithm receives a terminal that is drawn independently from a distribution over V , and needs to maintain a subset of edges connecting all the terminals received so far.

Our work differs from [20] in following four aspects. First, in the stochastic online optimization problem studied in [20], when each request i arrives, the previous requests $\{1, \dots, i-1\}$ have been realized, and the realization is known. By contrast, in the BGND problem, every agent i needs to be served without knowing the actual realization of the other agents. Second, the cost function studied in [20] maps each resource e to a fixed toll, which

is subadditive in the number of requests using e , while our cost function is superadditive. Third, in the BGND problem with the set connectivity requests, for each agent i , each type t_i is a set of terminals rather than a single terminal, and each action in $A_i^{t_i}$ is a Steiner tree spanning over the set of terminals corresponding to t_i . Fourth, in the BGND problem, each prior distribution p_i is over the types of agent i , while there is no distribution over the agents.

1.3 Paper Organization

The rest of this paper is organized as follows. Sec. 2 introduces some of the concepts employed in our approximation framework together with some notation and terminology. The main challenges that we had to overcome when developing this framework and some of the techniques used for that purpose are discussed in Sec. 3. Sec. 4 is dedicated to a detailed exposition of our approximation framework. Its performance is then analyzed in Sec. 5 using certain game theoretic properties.

2 Preliminaries

We follow the common convention that for an N -tuple $x = (x_1, \dots, x_N)$ and for $i \in [N]$, the notation x_{-i} denotes the $(N-1)$ -tuple $(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_N)$. Likewise, for a Cartesian product $X = X_1 \times \dots \times X_N$ and for $i \in [N]$, the notation X_{-i} denotes the Cartesian product $X_1 \times \dots \times X_{i-1} \times X_{i+1} \times \dots \times X_N$.

2.1 The BGND Game

Given an instance $\mathcal{I} = \langle N, E, \{T_i, p_i\}_{i \in [N]}, \{\xi_{e,j}\}_{e \in E, j \in [q]}, \{\alpha_j\}_{j \in [q]} \rangle$ of a BGND problem \mathcal{P} , we define a *BGND game* by associating every agent $i \in [N]$ with a strategic player who decides on the strategy s_i with the objective of minimizing her own individual cost defined as follows. Given an action profile $a \in A$ and a resource $e \in E$, the corresponding cost $F_e(l_e^a)$ is equally divided among the players $i \in [N]$ satisfying $e \in a_i$; in other words, the *cost share* of player i in resource e under action profile a , denoted by $f_{i,e}(a)$, is defined to be

$$f_{i,e}(a) = \begin{cases} 0, & e \notin a_i \\ \frac{F_e(l_e^a)}{|\{i: e \in a_i\}|} = \sum_j \xi_{e,j} (l_e^a)^{\alpha_j - 1}, & \text{otherwise} \end{cases}.$$

Informally, the individual cost of player i is the sum of her cost shares over all resources.

For a more formal treatment of the BGND game, we occasionally need to explicitly specify the type t_i of player i in the expressions involving her cost share in which case we use the notation $f_{i,e}(t_i; a)$, following the convention that $f_{i,e}(t_i; a) = f_{i,e}(a)$ if $a_i \in A_i^{t_i}$; and $f_{i,e}(t_i; a) = \infty$ otherwise. The individual cost of a player i with respect to the type t_i and a fixed action profile a is defined as $C_i(t_i; a) = \sum_{e \in E} f_{i,e}(t_i; a)$. Correspondingly, for each player $i \in [N]$ and each type $t_i \in T_i$, we define the *type-specified* expected individual cost $C_i(t_i; s) = \mathbb{E}_{t_{-i} \sim p_{-i}} [C_i(t_i; s(t_i, t_{-i}))]$. The objective function that player i wishes to minimize is her *type-averaged* expected individual cost $C_i(s) = \mathbb{E}_{t_i \sim p_i} [C_i(t_i; s)]$, irrespective of the total cost $C(s)$, often referred to as the *social cost*.

Let $f_{i,e}(a_i; s_{-i}) = \mathbb{E}_{t_{-i} \sim p_{-i}} [f_{i,e}(a_i, s_{-i}(t_{-i}))]$ be the expected cost share of player $i \in [N]$ on resource $e \in E$ with respect to action $a_i \in A_i$ and strategy profile $s_{-i} \in S_{-i}$. Fixing $a_{-i} \in A_{-i}$ (resp., $s_{-i} \in S_{-i}$), the cost share $f_{i,e}(a_i, a_{-i})$ (resp., expected cost share $f_{i,e}(a_i; s_{-i})$) of

player i on resource e is the same for every action $a_i \in A_i$ such that $e \in a_i$. Therefore, it is often convenient to ignore the specifics of action a_i and use the notations $f_{i,e}(+, a_{-i})$ and $\hat{f}_{i,e}(+; s_{-i})$ instead of $f_{i,e}(a_i, a_{-i})$ and $\hat{f}_{i,e}(a_i; s_{-i})$, respectively, given that $e \in a_i$.³

2.2 Definitions for the Algorithm Design and Analysis

The following definitions play key roles in the design and analysis of our framework.

► **Definition (Choice Function [32]).** A choice function $\sigma : T \mapsto A$ maps every type profile $t \in T$ to an action profile $a \in A^t$. The action specified by σ for player $i \in [N]$ with respect to type profile t is denoted by $\sigma_i(t)$. In particular, the choice function that maps each type profile t to an action profile that realizes $\text{OPT}(t)$ is denoted by σ^* .

► **Definition (Smoothness [32]).** Given parameters $\lambda > 0$ and $0 < \mu < 1$, a BGND game is said to be (λ, μ) -smooth if $\sum_{i \in [N]} C_i(t_i; (\sigma_i^*(t), a_{-i})) \leq \lambda \cdot \text{OPT}(t) + \mu \cdot \sum_{i \in [N]} C_i(t'_i, a)$ for every type profiles $t, t' \in T$ and action profile $a \in A^{t'}$.

► **Definition (Potential Function).** A function $\Phi : S \mapsto \mathbb{R}_{\geq 0}$ is said to be a potential function of the BGND game if $\Phi(s) - \Phi(s'_i, s_{-i}) = C_i(s) - C_i(s'_i, s_{-i})$ for every strategy profile $s \in S$, player $i \in [N]$, and strategy $s'_i \in S_i$. The potential function $\Phi(\cdot)$ is said to be K -bounded for a parameter $K \geq 1$ if $\Phi(s) \leq C(s) \leq K \cdot \Phi(s)$ for every strategy profile $s \in S$.

► **Definition ($(\underline{\eta}, \bar{\eta})$ -Estimation).** Given real parameters $\underline{\eta}, \bar{\eta} \geq 1$, a value x is said to be an $(\underline{\eta}, \bar{\eta})$ -estimation of the expected cost share $\hat{f}_{i,e}(a_i; s_{-i})$ (resp., $\hat{f}_{i,e}(+; s_{-i})$) if it satisfies $x/\underline{\eta} \leq \hat{f}_{i,e}(a_i; s_{-i}) \leq x \cdot \bar{\eta}$ (resp., $x/\underline{\eta} \leq \hat{f}_{i,e}(+; s_{-i}) \leq x \cdot \bar{\eta}$). We typically denote this estimation x by $\hat{f}_{i,e}(a_i; s_{-i})$ (resp., $\hat{f}_{i,e}(+; s_{-i})$). The BGND game is said to be poly-time $(\underline{\eta}, \bar{\eta})$ -estimable if for every player $i \in [N]$ and strategy profile $s_{-i} \in S_{-i}$, there exists an algorithm which runs in time $\text{poly}(N, q, |T_1|, \dots, |T_N|)$ and outputs an $(\underline{\eta}, \bar{\eta})$ -estimation of the expected cost share $\hat{f}_{i,e}(+; s_{-i})$. The BGND game is said to be tractable if it is poly-time $(\underline{\eta}, \bar{\eta})$ -estimable with $\bar{\eta} = \underline{\eta} = 1$.

Fix some player $i \in [N]$, type $t_i \in T_i$, and $(\underline{\eta}, \bar{\eta})$ -estimations $\hat{f}_{i,e}(s_i(t_i); s_{-i})$, $e \in E$. With respect to these variables, let $\hat{C}_i(t_i; s) = \sum_{e \in E} \hat{f}_{i,e}(s_i(t_i); s_{-i})$ and $\hat{C}_i(s) = \mathbb{E}_{t_i \sim p_i}[\hat{C}_i(t_i; s)]$. By the linearity of expectation, we know that $\hat{C}_i(t_i; s)/\underline{\eta} \leq C_i(t_i; s) \leq \hat{C}_i(t_i; s) \cdot \bar{\eta}$ and $\hat{C}_i(s)/\underline{\eta} \leq C_i(s) \leq \hat{C}_i(s) \cdot \bar{\eta}$. Consequently, we refer to $\hat{C}_i(t_i; s)$ and $\hat{C}_i(s)$ as $(\underline{\eta}, \bar{\eta})$ -estimations of $C_i(t_i; s)$ and $C_i(s)$, respectively.

► **Definition (Approximate Best Response).** For strategy profile $s \in S$ and player $i \in [N]$, strategy $s_i \in S_i$ is said to be an approximate best response (ABR) of i with approximation parameter $\chi \geq 1$ if $C_i(s_i, s_{-i}) \leq \chi \cdot C_i(s'_i, s_{-i})$ holds for any $s'_i \in S_i$. We may omit the explicit mention of the approximation parameter χ when it is clear from the context. A best response (BR) is an ABR with approximation parameter $\chi = 1$.

► **Definition (Approximate Best Response Dynamics).** An approximate best response dynamic (ABRD) is a procedure that starts from a predetermined strategy profile $s^0 \in S$ and generates a series of strategy profiles s^1, \dots, s^R such that for every $1 \leq r \leq R$, there exists some player $i \in [N]$ satisfying (1) $s_{-i}^r = s_{-i}^{r-1}$; and (2) s_i^r is an ABR of i to s_{-i}^{r-1} .

³ To avoid ambiguity concerning the definition of $f_{i,e}(+, a_{-i})$ and $\hat{f}_{i,e}(+; s_{-i})$ for resources $e \notin A_i$, we assume (in the scope of using these notations) that $A_i = E$ for all $i \in [N]$. This is without loss of generality as one can augment T_i with a virtual type \tilde{t}_i such that $A_i^{\tilde{t}_i} = \{E\}$ and $p_i(\tilde{t}_i)$ is arbitrarily small.

3 Overview of the Main Challenges and Techniques

The approximation framework presented in Sec. 4 for BGND problems is inspired by the framework designed in [15] for full information GND problems only in the conceptual sense that both algorithms employ approximate best response dynamics. In a high-level, for a certain number R of rounds that will be carefully chosen in order to achieve the approximation promise, and starting from some properly chosen initial strategy profile s^0 , for each round $1 \leq r \leq R$ the strategy profile s^r is generated from s^{r-1} in the following manner:

1. For every player $i \in [N]$ and resource $e \in E$, compute an $(\underline{\eta}, \bar{\eta})$ -estimation $\widehat{f}_{i,e}(+; s_{-i}^{r-1})$ of the expected cost share $f_{i,e}(+; s_{-i}^{r-1})$.
2. For every player $i \in [N]$, construct the strategy s'_i by mapping each type $t_i \in T_i$ to the action $a_i \in A_i^{t_i}$ computed by invoking the action ϱ -oracle with weight vector w defined by setting $w(e) = \widehat{f}_{i,e}(+; s_{-i}^{r-1})$.
3. Choose player $i \in [N]$ according to the game theoretic criterion presented in Sec. 4 regarding the estimations $\widehat{C}_i(s^{r-1})$ and $\widehat{C}_i(s'_i, s_{-i}^{r-1})$ of the type-averaged expected individual costs. Construct s^r by updating the strategy of the chosen player i to s'_i .

However, beyond the similar high-level structure, the technical construction in this paper is entirely different from [15] since the incomplete information assumption of the BGND setting exhibits new algorithmic challenges that require novel techniques. Specifically, the main challenges that our technical analysis in this paper handles are as follows.

A first obstacle here is the difficulty in computing the estimation $\widehat{f}_{i,e}(+; s_{-i}^{r-1}) = \mathbb{E}_{t_{-i} \sim p_{-i}}[f_{i,e}(+, s_{-i}(t_{-i}))]$ in step 1 since there are exponentially (in N) many possibilities for t_{-i} . Another source of difficulty in this regard is that the function $f_{i,e}(+, s_{-i}(t_{-i}))$ is nonlinear in $l_e^{s_{-i}(t_{-i})}$. One may hope that Jensen's inequality [26] can resolve this issue, however, as we explain in the technical sections, it is not enough for obtaining proper bounds on both $\underline{\eta}$ and $\bar{\eta}$. This obstacle is addressed in Sec. 5 (and Sec. 8 of the full version [16]) where we employ probabilistic tools from [8] and using Cantelli's inequality [34] to obtain the required estimation of the expression $\mathbb{E}_{t_{-i} \sim p_{-i}}[f_{i,e}(+, s_{-i}(t_{-i}))]$.

A second obstacle is that the ABRD-based approximation framework expresses its approximation guarantees in terms of smoothness parameters and bounded potential functions. However, neither the smoothness parameters nor the existence of a bounded potential function are known for the BGND game that we have defined here. We provide a new analysis for these two issues in Sec. 6 and Sec. 7 of the full version [16], respectively.

A third obstacle involves the stopping condition of the best response dynamics. A stopping condition for the full information case, via the smoothness framework, was developed by [33] (showing that if the current outcome in a best response dynamics is far from optimal there must exist a player whose best response significantly improves his own utility). For the Bayesian case, to the best of our knowledge, no such general stopping condition was known prior to the current paper. In fact, the smoothness framework for the Bayesian case which was developed in [32] did not include any results on best response dynamics. One specific technical difficulty is that Bayesian smoothness is defined in [32] w.r.t. a deviation to the optimal choice function rather than to a best response. This obstacle is resolved in Sec. 5 of the full version [16] where we provide such a stopping condition by proving that if the outcome of the current step of the ABRD in the Bayesian case is far from optimal, there must exist a player whose approximate best response must significantly improve her utility.

A fourth obstacle regards the output of the algorithm, once the ABRD terminates. Although we prove that there exists at least one strategy profile s^r , $1 \leq r \leq R$, with a sufficiently small social cost $C(s^r)$, we do not know how to find it. In particular, we wish to emphasize that we cannot simply evaluate the social cost function $C(\cdot)$ (see Eq. (2)) due

to the exponential number of type profiles. This obstacle does not exist in [15] where they can explicitly go over all steps of the full information ABRD and find the exact step whose outcome has minimal cost. To resolve this issue, we output the last strategy profile s^R generated in the ABRD and bound its loss. This is described in Sec. 5 of the full version [16].

Our technical constructions and our analysis employ various techniques from algorithmic game theory, demonstrating once again (as in [15]) the usefulness of this literature as a toolbox for algorithmic constructions that, on the face of it, have nothing to do with selfish agents. In particular, in this paper (and as assumed in the literature on oblivious routing [21, 17, 35]), we construct an algorithm that receives a correct input and outputs routing tables that the agents are going to follow without issues of selfish deviations.

4 The Algorithm

In this part, we present an algorithm, which is referred to as **Bayes-ABRD**, for a given BGND problem \mathcal{P} . The algorithm is assumed to have free access to an action ϱ -oracle for \mathcal{P} , which is denoted by $\mathcal{O}_{\mathcal{P}}$.

With an input instance $\mathcal{I} = \langle N, E, \{T_i, p_i\}_{i \in [N]}, \{\xi_{e,j}\}_{e \in E, j \in [q]}, \{\alpha_j\}_{j \in [q]} \rangle$, the first step of the algorithm is to (conceptually) construct a BGND game, and choose a tuple of parameters $(\lambda, \mu, K, \underline{\eta}, \bar{\eta})$ such that the BGND game

1. is (λ, μ) -smooth with $\varrho(\underline{\eta}\bar{\eta})^2\mu < 1$,
2. has a potential function Φ that is K -bounded,
3. is poly-time $(\underline{\eta}, \bar{\eta})$ -estimable.

The existence and exact values of the parameters in this tuple are presented in Sec. 5.

► **Lemma 3.** *For any $i \in [N]$ and any $s_{-i} \in S_{-i}$, there exists a $\text{poly}(|E|, N, q, \{|T_i|\}_{i \in [N]})$ -time procedure which generates a strategy $s_i \in S_i$ and the corresponding $(\underline{\eta}, \bar{\eta})$ -estimation $\hat{C}_i(s_i, s_{-i})$ of the individual costs such that $\hat{C}_i(s_i, s_{-i}) \leq \varrho \cdot \eta \cdot C_i(s'_i, s_{-i})$ for any $s'_i \in S_i$. This means in particular that s_i is an ABR of i to s_{-i} with approximation parameter $\varrho \cdot \underline{\eta}\bar{\eta}$.⁴*

Employing the procedure promised by Lemma 3, **Bayes-ABRD** simulates an ABRD of at most R rounds s^0, s^1, \dots for the BGND game induced by \mathcal{I} . Here R is a positive integer depending on the tuple $(\lambda, \mu, K, \underline{\eta}, \bar{\eta})$, and its exact value is also deferred to the following parts (Sec. 5). The ABRD simulated in our algorithm is done as follows.

Each player i chooses her initial strategy s_i^0 by taking each $s_i^0(t_i)$ to be the action generated by $\mathcal{O}_{\mathcal{P}}$ for type t_i with respect to the weight vector w^0 defined by setting $w^0(e) = \sum_{j \in [q]} \xi_{e,j}$, that is, as if i is playing alone. The obtained strategy s_i^0 is broadcast by player i to all the other players such that the full strategy profile s^0 is known by every player. Assuming that s^{r-1} , $1 \leq r \leq R$, was already constructed and known by all the players, s^r is obtained as follows. Every player $i \in [N]$ employs the procedure promised by Lemma 3 to generate an ABR \hat{s}_i^{r-1} to s_{-i}^{r-1} , and computes $\Delta_i^r = \hat{C}_i(s^{r-1}) - (\underline{\eta}\bar{\eta}) \cdot \hat{C}_i(\hat{s}_i^{r-1}, s_{-i}^{r-1})$. Both the strategy \hat{s}_i^{r-1} and the value Δ_i^r are broadcast to all the other players. If $\Delta_i^r \leq 0$ for all $i \in [N]$, then the ABRD stops, and every player i sets $s_i^r = s_i^{r-1}$; in this case, we say that the ABRD *converges*. Otherwise, fix $\Delta^r = \sum_{i \in [N]} \Delta_i^r$ and choose some player $i' \in [N]$ so that $\Delta_{i'}^r > 0$ and $\Delta_{i'}^r \geq \frac{1}{N}\Delta^r$ to update her strategy, setting $s^r = (\hat{s}_{i'}^{r-1}, s_{-i'}^{r-1})$ (the existence of such a player is guaranteed by the pigeonhole principle, and ties are always broken by choosing the player with the smallest index). Such an update can be performed by each

⁴ All subsequent occurrences of the term ABR (and ABRD) share the same approximation parameter $\varrho\underline{\eta}\bar{\eta}$, hence we may refrain from mentioning this parameter explicitly.

player in a distributed manner, as every player has the knowledge of the full vectors $\{s_i^r\}_{i \in [N]}$ and $\{\Delta_i^r\}_{i \in [N]}$. When the ABRD terminates (either because it has reached round $r = R$ or because it converges), **Bayes-ABRD** outputs the strategy generated in the last round.

► **Remark.** Note that **Bayes-ABRD** is designed for computing the strategy profile, not for invoking the strategies to decide the actions in real-time. All the operations of **Bayes-ABRD**, including broadcasting the strategy \hat{s}_i^{r-1} and the value Δ_i^r for every player i in every round $r \in [R]$, are carried out in a “precomputing stage” without seeing the realized type profile. The decision making that happens in real-time does not involve any further communication.

5 Analysis Sketch

Using the property of smoothness parameters, our analysis first gives the upper bound on the BCR with the tuple $(\lambda, \mu, K, \underline{\eta}, \bar{\eta})$ of parameters.⁵

► **Theorem 4.** Let $Q = \frac{2(\underline{\eta}\bar{\eta})N}{1-\varrho(\underline{\eta}\bar{\eta})^2\mu}$. If $R = \lceil Q \cdot \ln(KN^{\alpha_{\max}-1}) \rceil$, then the output s^{out} of **Bayes-ABRD** satisfies $C(s^{\text{out}}) \leq \frac{2K\varrho(\underline{\eta}\bar{\eta})^2\lambda}{1-\varrho(\underline{\eta}\bar{\eta})^2\mu} \cdot \mathbb{E}_{t \sim T}[\text{OPT}(t)]$.

Next, we consider the case where the parameters ϱ , $\underline{\eta}$ and $\bar{\eta}$ are fixed, and focus on finding proper smoothness parameters (λ, μ) such that the BGND game is (λ, μ) -smooth with $\mu < 1/[\varrho(\underline{\eta}\bar{\eta})^2]$. For any $\mu' \in (0, \frac{1}{\varrho(\underline{\eta}\bar{\eta})^2})$, define $g_{\mu'}(x) = (x+1)^{\alpha_{\max}-1} - \mu' \cdot x^{\alpha_{\max}}$ and $h(x) = \left[(\alpha_{\max}-1)(x+1)^{\alpha_{\max}-2} \right] / \left[\alpha_{\max} \cdot x^{\alpha_{\max}-1} \right]$. Define $\gamma_{z'}$ to be the unique positive root of $(x+1)^{z'-1} = x^{z'}$ for any $z' \geq 1$ [2]. Let $\mu_\alpha = h(\varrho(\underline{\eta}\bar{\eta})^2 \cdot \gamma_{\alpha_{\max}})$, and $\lambda_\alpha = g_{\mu_\alpha}(\varrho(\underline{\eta}\bar{\eta})^2 \cdot \gamma_{\alpha_{\max}})$. Then we have the following result on the smoothness.

► **Theorem 5.** The BGND game is $(\lambda_\alpha, \mu_\alpha)$ -smooth, and $\varrho(\underline{\eta}\bar{\eta})^2\mu_\alpha < 1 - 1/\alpha_{\max}$.

We then proceed to prove that the BGND game admits a potential function that is K -bounded with $K = \lceil \alpha_{\max} \rceil$.

► **Theorem 6.** For the BGND game, there exists a potential function $\Phi(s)$ that satisfies $\Phi(s) \leq C(s) \leq \lceil \alpha_{\max} \rceil \cdot \Phi(s)$ for any strategy profile s .

Now it remains to compute and analyze the $(\underline{\eta}, \bar{\eta})$ -estimation of the expected cost shares. For any $z \in (0, 1)$ and $z' \geq 1$, define $b_z = ((\beta^\circ)^2 + 1)(1 - \frac{1}{\beta^\circ})^{-z}$ with β° being the unique root of $2\beta^3 - (z+2)\beta^2 - 2 = 0$ in the interval $(1, +\infty)$, and $B_{z'}$ to be the fractional Bell number with the parameter z' [6, 28]. Our analysis utilizes the following propositions.

► **Lemma 7** ([8]). Let $\{X_1, X_2, \dots, X_k, \dots\}$ be a finite set of mutually independent random variables following the Bernoulli distribution supported on $\{0, 1\}$. Then for any $z \geq 1$, $\mathbb{E}\left[\left(\sum_k X_k\right)^z\right] \leq B_z \cdot \max\left\{\mathbb{E}\left[\sum_k X_k\right], \left(\mathbb{E}\left[\sum_k X_k\right]\right)^z\right\}$.

► **Lemma 8.** Let $\{X_1, X_2, \dots, X_k, \dots\}$ be a finite set of Bernoulli random variables that are mutually independent. For any $z' \in (0, 1)$, $\frac{1}{b_{z'}} \leq \mathbb{E}\left[\left(1 + \sum_k X_k\right)^{z'}\right] \leq \left(\mathbb{E}\left[1 + \sum_k X_k\right]\right)^{z'}$.

⁵ The proof of Theorem 4 bears similarity to the analysis in [33, 15]. Hence it is deferred to the attached full version. The main differences between that proof of Theorem [33, 15] are discussed in Sec. 3.

For each action a_i of player i and each resource e , denote the indicator of whether e is contained in a_i by $\delta(a_i, e)$. Formally,

$$\delta(a_i, e) = \begin{cases} 0 & \text{if } e \notin a_i \\ 1 & \text{otherwise} \end{cases}.$$

► **Theorem 9.** For any player i , any edge e , any action a_i , and any strategies s_{-i} , let

$$\widehat{f}_{i,e}(+; s_{-i}) = \sum_{j \in [q]} \xi_{e,j} \left[1 + \sum_{i' \neq i} \sum_{t_{i'} \in T_{i'}} p_{i'}(t_{i'}) \delta(s_{i'}(t_{i'}), e) \right]^{\alpha_j - 1}, \quad (4)$$

then $\frac{\widehat{f}_{i,e}(+; s_{-i})}{\max\{1, \max_{j: \alpha_j \in (1,2)} b_{\alpha_j - 1}\}} \leq f_{i,e}(+; s_{-i}) \leq \widehat{f}_{i,e}(+; s_{-i}) \cdot \{1, \max_{j: \alpha_j \geq 2} B_{\alpha_j - 1}\}$.

Sketch of Proof. Let a_i be an action in A_i satisfying $e \in a_i$. By definition, we have

$$\begin{aligned} f_{i,e}(+; s_{-i}) &= \mathbb{E}_{t_{-i} \sim p_{-i}} [f_{i,e}(a_i, s_{-i}(t_{-i}))] \\ &= \sum_{j \in [N]} \xi_{e,j} \cdot \mathbb{E}_{t_{-i} \sim p_{-i}} [(l_e^{a_i, s_{-i}(t_{-i})})^{\alpha_j - 1}] \\ &= \sum_{j \in [N]} \xi_{e,j} \cdot \mathbb{E}_{t_{-i} \sim p_{-i}} \left[\left(1 + \sum_{i' \in [N]: i' \neq i} \delta(s_{i'}(t_{-i}(i')), e) \right)^{\alpha_j - 1} \right] \\ &= \sum_{j \in [N]} \xi_{e,j} \mathbb{E}_{\{t_{i'} \sim p_{i'}\}_{i' \neq i}} \left[\left(1 + \sum_{i' \neq i} \delta(s_{i'}(t_{i'}), e) \right)^{\alpha_j - 1} \right]. \end{aligned}$$

The last transition holds because the prior distribution p is assumed to be a product distribution.

Now define a finite set of mutually independent Bernoulli random variables $\{X_{i',e}(s)\}_{i' \neq i}$ such that each $X_{i',e}(s)$ takes the value 1 with probability $\sum_{t_{i'}: e \in s_{i'}(t_{i'})} p_{i'}(t_{i'})$. Then it can be inductively proved that

$$\mathbb{E}_{\{t_{i'} \sim p_{i'}\}_{i' \neq i}} \left[\left(1 + \sum_{i' \neq i} \delta(s_{i'}(t_{i'}), e) \right)^{\alpha_j - 1} \right] = \mathbb{E} \left[\left(1 + \sum_{i' \neq i} X_{i',e}(s) \right)^{\alpha_j - 1} \right].$$

Recall that the constant 1 in the last expression above can also be viewed as a Bernoulli random variable which equals to 1 with probability 1. For every $\alpha_j \geq 2$, Lemma 7 gives

$$\begin{aligned} \mathbb{E} \left[\left(1 + \sum_{i' \neq i} X_{i',e}(s) \right)^{\alpha_j - 1} \right] &\leq B_{\alpha_j - 1} \cdot \max \left\{ \mathbb{E} \left[1 + \sum_{i' \neq i} X_{i',e}(s) \right], \left(\mathbb{E} \left[1 + \sum_{i' \neq i} X_{i',e}(s) \right] \right)^{\alpha_j - 1} \right\} \\ &= B_{\alpha_j - 1} \cdot \left(\mathbb{E} \left[1 + \sum_{i' \neq i} X_{i',e}(s) \right] \right)^{\alpha_j - 1}. \end{aligned}$$

The second line holds because $\mathbb{E} \left[1 + \sum_{i' \neq i} X_{i',e}(s) \right] > 1$. Similarly, it can be derived from Lemma 8 that for every $\alpha_j \in (1, 2)$,

$$\mathbb{E} \left[\left(1 + \sum_{i' \neq i} X_{i',e}(s) \right)^{\alpha_j - 1} \right] \leq \left(\mathbb{E} \left[1 + \sum_{i' \neq i} X_{i',e}(s) \right] \right)^{\alpha_j - 1},$$

which also trivially holds for $\alpha_j = 1$. So, $\mathbb{E} \left[\left(1 + \sum_{i' \neq i} X_{i',e}(s) \right)^{\alpha_j - 1} \right] \leq \max \left\{ 1, \max_{j: \alpha_j \geq 2} B_{\alpha_j - 1} \right\} \left(\mathbb{E} \left[1 + \sum_{i' \neq i} X_{i',e}(s) \right] \right)^{\alpha_j - 1}$, and in a similar way, it also be

inferred from Lemma 7 and Lemma 8 that $\mathbb{E}\left[\left(1 + \sum_{i' \neq i} X_{i',e}(s)\right)^{\alpha_j - 1}\right] \geq \left(\mathbb{E}\left[1 + \sum_{i' \neq i} X_{i',e}(s)\right]\right)^{\alpha_j - 1} / \max\{1, \max_{j: \alpha_j < 2} b_{\alpha_j - 1}\}$. Since $\mathbb{E}\left[1 + \sum_{i' \neq i} X_{i',e}(s)\right] = 1 + \sum_{i' \neq i} \sum_{t_{i'}} p_{i'}(t_{i'}) \delta(s_{i'}(t_{i'}), e)$, this proposition holds. \blacktriangleleft

Theorem 9 shows that for any $i \in [N]$, $e \in E$ and any s_{-i} , there exists a $(\max\{1, \max_{\alpha_j \in (1,2)} b_{\alpha_j - 1}\}, \max\{\max_{\alpha_j \geq 2} B_{\alpha_j - 1}, 1\})$ -estimation $\widehat{f}_{i,e}(+; s_{-i})$ of $f_{i,e}(+; s_{-i})$, and the following proposition indicates that such an estimation can be obtained in $\text{poly}(q, N, \{|T_i|\}_{i \in [N]})$ -time.

► **Corollary 10.** *By computing Eq. (4), the desired estimation of each expected cost share is obtained in $O(q \cdot \sum_{i \in [N]} |T_i|)$ -time.*

Plugging Theorem 5, Theorem 6, Theorem 9, and Corollary 10 into Theorem 4 proves our main result, Theorem 2.

References

- 1 Ajit Agrawal, Philip Klein, and R. Ravi. When Trees Collide: An Approximation Algorithm for the Generalized Steiner Problem on Networks. *SIAM Journal on Computing*, 24(3):440–456, 1995.
- 2 Sebastian Aland, Dominic Dumrauf, Martin Gairing, Burkhard Monien, and Florian Schoppmann. Exact price of anarchy for polynomial congestion games. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 218–229. Springer, 2006.
- 3 Susanne Albers. Energy-efficient Algorithms. *Commun. ACM*, 53(5):86–96, 2010.
- 4 Noga Alon, Yuval Emek, Michal Feldman, and Moshe Tennenholtz. Bayesian ignorance. *Theoretical Computer Science*, 452:1–11, 2012. Preliminary version appears in Proceedings of the 29th ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, pp. 384–391. ACM, 2010.
- 5 Matthew Andrews, Antonio Fernández Anta, Lisa Zhang, and Wenbo Zhao. Routing for Power Minimization in the Speed Scaling Model. *IEEE/ACM Transactions on Networking*, 20(1):285–294, February 2012.
- 6 Evripidis Bampis, Alexander Kononov, Dimitrios Letsios, Giorgio Lucarelli, and Maxim Sviridenko. Energy efficient scheduling and routing via randomized rounding. In *33rd International Conference on Foundations of Software Technology and Theoretical Computer Science*, page 449, 2013.
- 7 Nikhil Bansal, Tracy Kimbrel, and Kirk Pruhs. Speed Scaling to Manage Energy and Temperature. *J. ACM*, 54(1):3:1–3:39, 2007.
- 8 Daniel Berend and Tamir Tassa. Improved bounds on Bell numbers and on moments of sums of random variables. *Probability and Mathematical Statistics*, 30(2):185–205, 2010.
- 9 Jarosław Byrka, Fabrizio Grandoni, Thomas Rothvoss, and Laura Sanità. Steiner tree approximation via iterative randomized rounding. *Journal of the ACM (JACM)*, 60(1):6, 2013. Preliminary version in STOC'10.
- 10 Moses Charikar, Chandra Chekuri, To-yat Cheung, Zuo Dai, Ashish Goel, Sudipto Guha, and Ming Li. Approximation Algorithms for Directed Steiner Problems. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 192–200, 1998.
- 11 Chandra Chekuri, Guy Even, Anupam Gupta, and Danny Segev. Set Connectivity Problems in Undirected Graphs and the Directed Steiner Network Problem. *ACM Trans. Algorithms*, 7(2):18:1–18:17, 2011.
- 12 Ken Christensen, Pedro Reviriego, Bruce Nordman, Michael Bennett, Mehrgan Mostowfi, and Juan Antonio Maestro. IEEE 802.3 az: the road to energy efficient ethernet. *IEEE Communications Magazine*, 48(11), 2010.

- 13 Teodor Gabriel Crainic, Xiaorui Fu, Michel Gendreau, Walter Rei, and Stein W Wallace. Progressive hedging-based metaheuristics for stochastic network design. *Networks*, 58(2):114–124, 2011.
- 14 E. W. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numer. Math.*, 1(1):269–271, 1959.
- 15 Yuval Emek, Shay Kutten, Ron Lavi, and Yangguang Shi. Approximating Generalized Network Design Under (Dis)Economies of Scale with Applications to Energy Efficiency. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, pages 598–606, New York, NY, USA, 2018. ACM.
- 16 Yuval Emek, Shay Kutten, Ron Lavi, and Yangguang Shi. Bayesian Generalized Network Design. *ArXiv e-prints*, June 2019. [arXiv:1907.00484](https://arxiv.org/abs/1907.00484).
- 17 Matthias Englert and Harald Räcke. Oblivious Routing for the Lp-norm. In *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '09, pages 32–40, Washington, DC, USA, 2009. IEEE Computer Society.
- 18 Matthias Englert and Harald Räcke. Oblivious routing for the Lp-norm. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 32–40. IEEE, 2009.
- 19 Michael L. Fredman and Robert Endre Tarjan. Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms. *J. ACM*, 34(3):596–615, 1987.
- 20 Naveen Garg, Anupam Gupta, Stefano Leonardi, and Piotr Sankowski. Stochastic analyses for online combinatorial optimization problems. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '08, pages 942–951. Society for Industrial and Applied Mathematics, 2008.
- 21 Anupam Gupta, Mohammad T Hajiaghayi, and Harald Räcke. Oblivious network design. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 970–979. Society for Industrial and Applied Mathematics, 2006.
- 22 Anupam Gupta, R Ravi, and Amitabh Sinha. An edge in time saves nine: LP rounding approximation algorithms for stochastic network design. In *FOCS*, pages 218–227, 2004.
- 23 Prahladh Harsha, Thomas P. Hayes, Hariharan Narayanan, Harald Räcke, and Jaikumar Radhakrishnan. Minimizing Average Latency in Oblivious Routing. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '08, pages 200–207, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.
- 24 Intel. Enhanced Intel Speedstep Technology for the Intel Pentium M Processor. In *Intel White Paper 301170-001*, 2004.
- 25 Sandy Irani and Kirk R. Pruhs. Algorithmic Problems in Power Management. *SIGACT News*, 36(2):63–76, 2005.
- 26 J.L.W.V. Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(1):175–193, 1906.
- 27 Gregory Lawler and Hariharan Narayanan. Mixing Times and ϵ -P Bounds for Oblivious Routing. In *Proceedings of the Meeting on Analytic Algorithmics and Combinatorics*, ANALCO '09, pages 66–74, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics.
- 28 Konstantin Makarychev and Maxim Sviridenko. Solving Optimization Problems with Dis-economies of Scale via Decoupling. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, 2014.
- 29 Sergiu Nedeveschi, Lucian Popa, Gianluca Iannaccone, Sylvia Ratnasamy, and David Wetherall. Reducing Network Energy Consumption via Sleeping and Rate-adaptation. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, NSDI'08, pages 323–336. USENIX Association, 2008.
- 30 Harald Räcke. Survey on Oblivious Routing Strategies. In *Mathematical Theory and Computational Practice*, volume 5635 of *Lecture Notes in Computer Science*, pages 419–429. Springer Berlin Heidelberg, 2009.

45:16 Bayesian Generalized Network Design

- 31 Ragheb Rahmaniani, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei. Accelerating the Benders Decomposition Method: Application to Stochastic Network Design Problems. *SIAM Journal on Optimization*, 28(1):875–903, 2018.
- 32 Tim Roughgarden. The price of anarchy in games of incomplete information. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 862–879. ACM, 2012.
- 33 Tim Roughgarden. Intrinsic robustness of the price of anarchy. *Journal of the ACM (JACM)*, 62(5):32, 2015. Preliminary version in STOC’09.
- 34 I Richard Savage. Probability inequalities of the Tehebycheff type. *Journal of Research of the National Bureau of Standards-B. Mathematics and Mathematical Physics B*, 65(3):211–222, 1961.
- 35 Yangguang Shi, Fa Zhang, Jie Wu, and Zhiyong Liu. Randomized oblivious integral routing for minimizing power cost. *Theoretical Computer Science*, 607:221–246, 2015.
- 36 Adam Wierman, Lachlan LH Andrew, and Ao Tang. Power-Aware Speed Scaling in Processor Sharing Systems. In *28th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, INFOCOM*, pages 2007–2015, April 2009.
- 37 F. Yao, A. Demers, and S. Shenker. A scheduling model for reduced CPU energy. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 374–382, 1995.