Best Arm Identification in Graphical Bilinear Bandits

Geovani Rizk, Thomas Albert, Igor Colin, Rida Laraki, Yann Chevaleyre

To cite this version:

Geovani Rizk, Thomas Albert, Igor Colin, Rida Laraki, Yann Chevaleyre. Best Arm Identification in Graphical Bilinear Bandits. ICML 2021 - Proceedings of the 38th International Conference on Machine Learning, Jul 2021, Pittsburgh, United States. hal-03413102

HAL Id: hal-03413102

https://hal.inria.fr/hal-03413102

Submitted on 5 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

arXiv:2012.07641v3 [cs.LG] 10 Jun 2021

Best Arm Identification in Graphical Bilinear Bandits

Geovani Rizk [1 2] Albert Thomas [2] Igor Colin [2] Rida Laraki [1 3] Yann Chevaleyre [1]

Abstract

We introduce a new graphical bilinear bandit problem where a learner (or a *central entity*) allocates arms to the nodes of a graph and observes for each edge a noisy bilinear reward representing the interaction between the two end nodes. We study the best arm identification problem in which the learner wants to find the graph allocation maximizing the sum of the bilinear rewards. By efficiently exploiting the geometry of this bandit problem, we propose a *decentralized* allocation strategy based on random sampling with theoretical guarantees. In particular, we characterize the influence of the graph structure (e.g. star, complete or circle) on the convergence rate and propose empirical experiments that confirm this dependency.

1. Introduction

In many multi-agent systems the contribution of an agent to a common team objective is impacted by the behavior of the other agents. The agents must coordinate (or be coordinated) to achieve the best team performance. Consider, for instance, the problem of configuring antennas of a wireless cellular network to obtain the best signal quality over the whole network (Siomina et al., 2006). The signal quality of the region covered by a given antenna might be degraded by the behavior of its neighboring antennas due to an increase of interferences or bad user handovers. Another example is the adjustment of the turbine blades of a wind farm where the best adjustment for one turbine may generate turbulence for its neighboring turbines and thus be suboptimal for the global wind farm objective (Bargiacchi et al., 2018).

These real-life problems can be viewed as instances of a *stochastic multi-agent multi-armed bandit* problem (Robbins, 1952; Bargiacchi et al., 2018) where a learner (or a *central entity*) sequentially pulls a joint arm, one arm for each agent (*e.g.,* all the configuration parameters of the antennas), and receives an associated global noisy reward (*e.g.,* the signal quality over the whole network). The goal of the learner can either be to maximize the accumulated reward, implying a trade-off between exploration and exploitation, or to find the joint arm maximizing the reward, known as *pure exploration* or *best arm identification* (Bubeck et al., 2009; Audibert and Bubeck, 2010).

In this paper we focus on the best arm identification problem in a multi-agent system for which we assume the knowledge of a coordination graph $\mathcal{G} = (V, E)$ representing the agent interactions (Guestrin et al., 2002).

At each round t, a learner

1. chooses for each node $i \in V$ an arm $x_t^{(i)}$ in an finite arm set $\mathcal{X} \subset \mathbb{R}^d$,
2. observes for each edge $(i, j) \in E$ a bilinear reward $r_t^{(i,j)} = x_t^{(i)\top} \mathbf{M}_\star x_t^{(j)} + \eta_t^{(i,j)}$.

Here, we denote by $\mathbf{M}_\star \in \mathbb{R}^{d\times d}$ the unknown parameter matrix, and $\eta_t^{(i,j)}$ a zero-mean σ-sub-Gaussian random variable for all edges $(i, j) \in E$ and round t.

The goal of the central entity is to find, within a minimum number of rounds, the joint arm $(x_\star^{(1)}, \dots, x_\star^{(|V|)})$ such that the expected global reward $\sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)}$ is maximized.

The reward $r_t^{(i,j)}$ reflects the quality of the interaction between the neighboring nodes i and j when pulling respectively the arm $x_t^{(i)}$ and $x_t^{(j)}$ at time t. For instance, when configuring handover parameters of a wireless network, $r_t^{(i,j)}$ can be any criterion assessing the handover quality between antenna i and antenna j, the parameters selected by each antenna both impacting this quantity. The bilinear setting appears as a natural extension of the commonly studied linear setting to model the interaction between two agents. Furthermore, instead of a global reward being a sum of independent linear agent rewards, the global reward is now the result of the interactions between neighboring agents.

As exposed in Jun et al. (2019), the bilinear reward can be

[1] PSL - Université Paris Dauphine, CNRS, LAMSADE, Paris, France [2] Huawei Noah's Ark Lab [3] Liverpool University. Correspondence to: Geovani Rizk <geovani.rizk@dauphine.psl.eu>.

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

written as a linear reward in a higher dimensional space:

$$r_t^{(i,j)} = \mathrm{vec}\left(x_t^{(i)} x_t^{(j)\top}\right)^\top \mathrm{vec}\left(\mathbf{M}_\star\right) + \eta_t^{(i,j)} \quad , \qquad (1)$$

where for any matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$, $\mathrm{vec}\left(\mathbf{A}\right)$ denotes the vector in $\mathbb{R}^{d^2}$ which is the concatenation of all the columns of $\mathbf{A}$.

Since the unknown parameter $\mathbf{M}_\star$ is common to all the edges (i,j) of the graph, the expected global reward at time t can also be written as the scalar product $\left\langle \sum_{(i,j)\in E} \mathrm{vec}\left(x_t^{(i)} x_t^{(j)\top}\right), \mathrm{vec}\left(\mathbf{M}_\star\right)\right\rangle$. Hence, solving the best arm identification problem in the described graphical bilinear bandit boils down to solving the same problem in a global linear bandit. Although this trick allows to use classical algorithms in linear bandits, the number of joint arms is growing exponentially with the number of nodes, making such methods impractical.

Another possible way to address this problem based on equation (1) is to consider one linear bandit per edge, with constraints between edges. For more clarity, let us define the arm set $\mathcal{Z} = \{\mathrm{vec}\left(xx'^\top\right) \mid (x,x') \in \mathcal{X}^2\}$, and let us refer to any $z \in \mathcal{Z}$ as an *edge-arm* and to any $x \in \mathcal{X}$ as an *node-arm*. At each round t, the learner chooses for each edge (i,j) an edge-arm in $\mathcal{Z}$ with the constraint that for any pair of edges (i,j) and (i,k), if the edge-arm $\mathrm{vec}\left(xx'^\top\right)$ is assigned to the edge (i,j) and the edge-arm $\mathrm{vec}\left(x''x'''^\top\right)$ is assigned to the edge (i,k), then it must be that $x = x''$.

Given this constraint, how do we choose the appropriate sequence of edge-arms in order to build a good estimate of $\mathrm{vec}\left(\mathbf{M}_\star\right)$? Moreover, assuming we have built such a good estimator, is there a tractable algorithm to identify the best joint arm, or at least to find a joint arm yielding a high expected reward? In this paper, we answer these questions and provide algorithms and theoretical guarantees.

We show that even with a perfect estimator $\mathrm{vec}(\hat{\mathbf{M}}) = \mathrm{vec}\left(\mathbf{M}_\star\right)$, identifying the best joint arm is NP-Hard. To address this issue, we design a polynomial time twofold algorithm. Given $\mathrm{vec}(\hat{\mathbf{M}})$, it first identifies the best edge-arm $z_\star \in \mathcal{Z}$ maximizing $\langle z_\star, \mathrm{vec}(\hat{\mathbf{M}})\rangle$. Then, it allocates $z_\star$ to a carefully chosen subset of edges. We show that this yields a good approximation ratio in Section 4.

To build our estimator $\mathrm{vec}(\hat{\mathbf{M}})$, we rely on the G-Allocation strategy, as in Soare et al. (2014). We show that there exists a sampling procedure over the node-arms such that the associated edge-arms follow the optimal G-allocation strategy developed in the linear bandit literature. This procedure allows us to avoid the difficulty of having to satisfy the edge-arm constraints explicitly. Furthermore, we analyze the sample complexity of this method. This is detailed in Section 5.

In addition, we highlight the impact of the graph structure in Section 6 and provide the explicit repercussion on the convergence rate of the algorithm for different types: star, complete, circle and matching graphs. In particular, we show that for favorable graph structures (e.g. circles), our convergence rate matches that of standard linear bandits. Finally, Section 7 evidences the theoretical findings on numerical experiments.

2. Related Work

Best arm identification in linear bandits. There exists a vast literature on the problem of best arm identification in linear bandits (Soare et al., 2014; Xu et al., 2018; Degenne et al., 2020; Kazerouni and Wein, 2019; Zaki et al., 2020; Jedra and Proutiere, 2020), would it be by using greedy strategies (Soare et al., 2014), rounding procedures (Fiez et al., 2019) or random sampling (Tao et al., 2018). Although our problem can be formulated as a linear bandit problem, none of the existing methods would scale-up with the number of agents. Nevertheless, we will be relying on classical techniques, and more specifically those developed in Soare et al. (2014).

Bilinear bandits. Bandits with bilinear rewards have been studied in Jun et al. (2019). The authors derived a no-regret algorithm based on Optimism in the Face of Uncertainty Linear bandit (OFUL) (Abbasi-Yadkori et al., 2011), using the fact that a bilinear reward can be expressed as a linear reward in higher dimension. Our work extends their setting by considering a set of dependent bilinear bandits. Besides, the goal here is to find the best arm rather than minimizing the regret.

Bandits and graphs. Graphs are often used to bring structure to a bandit problem. In Valko et al. (2014) and Mannor and Shamir (2011), the arms are the nodes of a graph and pulling an arm gives information on the rewards of the neighboring arms. The reader can also refer to Valko (2020) for an account on such problems. In Cesa-Bianchi et al. (2013) each node is an instance of a linear bandit and the neighboring nodes are assumed to have similar unknown regression coefficients. The main difference with our setting is that the rewards of the nodes are independent.

Combinatorial and multi-agent bandits. Allocating arms to each node of a graph to then observe a global reward is a combinatorial bandit problem (Cesa-Bianchi and Lugosi, 2012), the number of joint arms scaling exponentially with the number of nodes. This has been extensively studied both in the regret-based (Chen et al., 2013; Perrault et al., 2020) and the pure exploration context (Chen et al., 2014; Cao and Krishnamurthy, 2019; Jourdan et al., 2021; Du et al., 2020). Our problem is closer to the one presented in Amin et al. (2011) and Bargiacchi et al. (2018), where several agents want to maximize a global team reward that can be decomposed into a sum of observable local rewards as in a *semi-bandit game* (Audibert et al., 2011; Chen et al., 2013). However, we study a more structured context as we assume observable bilinear rewards for each edge of the graph. Furthermore, note that our problem can be solved by the algorithm presented in Du et al. (2020) with a sample complexity increasing in the number of nodes. On the contrary, we propose in this paper an algorithm with a sample complexity decreasing in the number of nodes exploiting the structure of the bilinear reward and the graph. Finally, most of the algorithms developed for combinatorial bandits assume the availability of an oracle to solve the combinatorial optimization problem returning the arm to play or the final best arm recommendation. We make no such assumption.

3. Preliminaries and Notations

Let $\mathcal{G} = (V, E)$ be a directed graph with V the set of nodes, E the set of edges where we assume that if $(i, j) \in E$ then $(j, i) \in E$, and $\mathcal{N}(i)$ the set containing the neighbors of a node $i \in V$. We denote by $n = |V|$ the number of nodes and $m = |E|$ the number of edges. We define the *graphical bilinear bandit* on the graph $\mathcal{G}$ as the setting where a learner sequentially pulls at each round t a joint arm $\left(x_t^{(1)}, \dots, x_t^{(n)}\right) \in \mathcal{X}^n$, also called graph allocation or simply allocation when it is clear from the context, and then receives a bilinear reward $r_t^{(i,j)}$ for each edge $(i, j) \in E$. At each round, the joint arm can be constructed simultaneously or sequentially, however all the bilinear rewards are only revealed after the joint arm has been pulled.

We denote $K = |\mathcal{X}|$ the number of node-arms and it is assumed that $\mathcal{X}$ spans $\mathbb{R}^d$. For each round t of the learning procedure and each node $i \in V$, $x_t^{(i)} \in \mathcal{X}$ represents the node-arm allocated to the node $i \in V$. For each edge $(i, j) \in E$, we denote $z_t^{(i,j)} = \text{vec}\,(x_t^{(i)} x_t^{(j)\top}) \in \mathcal{Z}$ the associated chosen edge-arm.

The goal is to derive an algorithm that minimizes the number of pulled joint arms required to find the one maximizing the sum of the associated expected bilinear rewards, for a given confidence level. For the sake of simplicity, we assume that the unknown parameter matrix $\mathbf{M}_\star$ in the bilinear reward is symmetric. We provide an analysis of the non-symmetric case in Appendix E.

For any finite set X, $\mathcal{S}_X \triangleq \{\lambda \in [0, 1]^{|X|}, \sum_{x \in X} \lambda_x = 1\}$ denotes the simplex in $\mathbb{R}^{|X|}$. For any vector $x \in \mathbb{R}^d$, $\|x\|$ will denote the ℓ_2-norm of x. For any square matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$, we denote by $\|\mathbf{A}\| \triangleq \sup_{x:\|x\|=1} \|\mathbf{A}x\|$ the spectral norm of $\mathbf{A}$. Finally, for any vector $x \in \mathbb{R}^d$ and a symmetric positive-definite matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$, we define $\|x\|_{\mathbf{A}} \triangleq \sqrt{x^\top \mathbf{A} x}$.

4. An NP-Hard Problem

In this section, we address the problem of finding the best joint arm given $\mathbf{M}_\star$ or a good estimator $\hat{\mathbf{M}}$. If the best edge-arm $z_\star$ is composed of a single node-arm $x_\star$, that is $z_\star = \text{vec}\,(x_\star x_\star^\top)$, then finding the best joint arm is trivial and the solution is to assign $x_\star$ to all nodes. Conversely, if $z_\star$ is composed of two distinct node-arms $(x_\star, x'_\star)$, the problem is harder.

The following theorem states that, even with the knowledge of the true parameter $\mathbf{M}_\star$, identifying the best join-arm is NP-Hard with respect to the number of nodes n.

Theorem 4.1. *Consider a given matrix $\mathbf{M}_\star \in \mathbb{R}^{d\times d}$ and a finite arm set $\mathcal{X} \subset \mathbb{R}^d$. Unless P=NP, there is no polynomial time algorithm guaranteed to find the optimal solution of*

$$\max_{\left(x^{(1)},\dots,x^{(n)}\right)\in\mathcal{X}^n} \sum_{(i,j)\in E} x^{(i)\top} \mathbf{M}_\star \, x^{(j)} \ .$$

The proof of this theorem is in Appendix A and relies on a reduction to the Max-Cut problem. Hence, no matter which estimate $\hat{\mathbf{M}}$ of $\mathbf{M}_\star$ one can build, the learner is not guaranteed to find in polynomial time the joint arm $\left(x_\star^{(1)}, \dots, x_\star^{(n)}\right)$ maximizing the expected global reward. However, one can notice that, given the matrix $\mathbf{M}_\star$ or even a good enough estimate $\hat{\mathbf{M}}$, identifying the edge-arm $z^\star = \text{vec}\,(x_\star x_\star'^\top) \in \mathcal{Z}$ that maximizes the reward $z_\star^\top \,\text{vec}\,(\mathbf{M}_\star)$ requires only K^2 reward estimations (we simply estimate all the linear reward associated to each edge-arm in $\mathcal{Z}$). Thus, instead of looking for the best joint arm explicitely, we will first identify the best edge-arm $z^\star$, and then allocate $z^\star$ to the largest number of edges in the graph. We will also show that this approach gives a guarantee on its associated global reward.

Let us consider the graph allocation that places the maximum number of edge-arms $z_\star$ in $\mathcal{G}$. It is easy to show that the subgraph containing only the edges where $z_\star$ has been pulled is the largest bipartite subgraph included in $\mathcal{G}$. Recall that a graph $\mathcal{G}' = (V', E')$ is a bipartite if and only if one can partition the node set $V' = (V'_1, V'_2)$ such that

$$(i, j) \in E' \Rightarrow (i, j) \in V'_1 \times V'_2 \text{ or } (j, i) \in V'_1 \times V'_2 \ .$$

Notice, that if $\mathcal{G}'$ is the largest bipartite subgraph in $\mathcal{G}$, the number of edges in E' is the maximal number of edge-arms $z_\star$ that can be allocated with a single graph allocation.

Hence, finding the joint arm with the largest number of edge-arms $z_\star$ allocated in the graph is equivalent to finding the largest bipartite subgraph $\mathcal{G}' = (V', E')$ in $\mathcal{G}$. Once that subgraph is determined, we just need to allocate to all the nodes in V'_1 the node-arm $x_\star$ and to all the nodes of V'_2 the node-arm $x'_\star$ (which is equivalent to allocating to all the edges in E' the edge-arm $z_\star$).

Furthermore, we know that every m-edge graph contains a bipartite subgraph of at least $m/2$ edges (Erdos, 1975). Therefore, we propose Algorithm 1 which iteratively constructs a bipartite subgraph and allocates the nodes accordingly to create at least $m/2$ edge-arms $z_\star$.

Algorithm 1 Bipartite graph algorithm for Best Arm Identification in Graphical Bilinear Bandits

```
Input   :$\mathcal{G} = (V, E)$, $\mathcal{X}$, $\mathbf{M}$
Find $(x_\star, x'_\star) \in \arg\max_{(x,x')\in\mathcal{X}^2} x^\top \mathbf{M}\, x'$
Set $V_1 = \emptyset$, $V_2 = \emptyset$
for $i$ in $V$ do
    Set $n_1$ the number of neighbors of $i$ in $V_1$
    Set $n_2$ the number of neighbors of $i$ in $V_2$
    if $n_1 > n_2$ then
        $x^{(i)} = x'_\star$
        $V_2 \leftarrow V_2 \cup \{i\}$
    else
        $x^{(i)} = x_\star$
        $V_1 \leftarrow V_1 \cup \{i\}$
    end
end
return $\mathbf{x} = \left(x^{(1)}, \dots, x^{(n)}\right)$
```

The following result gives the guarantee on the global reward associated to the joint arm returned by Algorithm 1. We refer the reader to Appendix A for the proof.

Theorem 4.2. *Let us consider the graph $\mathcal{G} = (V, E)$, a finite arm set $\mathcal{X} \subset \mathbb{R}^d$ and the matrix $\mathbf{M}_\star$ given as input to Algorithm 1. Then, the expected global reward $r = \sum_{(i,j)\in E} x^{(i)\top}\mathbf{M}_\star x^{(j)}$ associated to the returned allocation $\mathbf{x} = \left(x^{(1)}, \dots, x^{(n)}\right) \in \mathcal{X}^n$ verifies:*

$$\frac{r - r_{\min}}{r_\star - r_{\min}} \geq \frac{1}{2} \ .$$

where $r_\star$ and $r_{\min}$ are respectively the highest and lowest global reward one can obtain with the appropriate joint arm. Finally, the complexity of the algorithm is in $\mathcal{O}(K^2 + n)$.

This type of approximation result is sometimes referred to as *differential approximation* or *z-approximation*, and is often viewed as a more subtle analysis than standard approximation ratio. We emphasize that finding a better ratio than $\frac{1}{2}$ is a very hard task: such a finding would immediately yield an improved differential approximation ratio for the Max-Cut problem, which is an opened problem since 2001 (Hassin and Khuller, 2001).

5. Construction of the Estimate $\hat{\mathbf{M}}$

In the previous section, we designed a polynomial time method that computes a $1/2$-approximation to the NP-Hard problem of finding the best joint arm given $\mathbf{M}_\star$. Notice that $\mathbf{M}_\star$ is only used to identify the best edge-arm $z_\star$. Thus, using an estimate $\hat{\mathbf{M}}$ of $\mathbf{M}_\star$ having the following property:

$$\arg\max_{z\in\mathcal{Z}} z^\top \operatorname{vec}(\hat{\mathbf{M}}) = \arg\max_{z\in\mathcal{Z}} z^\top \operatorname{vec}(\mathbf{M}_\star) \ , \qquad (2)$$

would still allow us to identify $z_\star$, and would thus give us the same guarantees.

In this section we tackle the problem of pulling the edge-arms during the learning procedure such that the estimated unknown parameter verifies (2) in as few iterations as possible. To do so, we first formalize the objective related to the linearized version of the problem. Then, we propose an algorithm reaching the given objective with high probability while satisfying the edge-arms constraints.

We denote by $\theta_\star = \operatorname{vec}(\mathbf{M}_\star)$ the parameter of the linearized problem and $\hat{\theta}_t$ the Ordinary Least Squares (OLS) estimate of $\theta_\star$ computed with all the data collected up to round t. The empirical gap between two edge-arms z and z' in $\mathcal{Z}$ is denoted $\hat{\Delta}_t(z, z') \triangleq (z - z')^\top \hat{\theta}_t$.

5.1. A Constrained G-Allocation

The goal here is to define the optimal sequence $(z_1, \dots, z_{mt}) \in \mathcal{Z}^{mt}$ that should be pulled in the first t rounds so that (2) is reached as soon as possible. A natural approach is to rely on classical strategies developed for best arm identification in linear bandits. Most of the known strategies (see *e.g.,* Soare et al. (2014); Xu et al. (2018); Fiez et al. (2019)) are based on a bound of the gap error $|(\theta_\star - \hat{\theta}_t)^\top(z - z')|$ for all $z, z' \in \mathcal{Z}$. This bound is then used to derive a stopping condition, indicating a sufficient number of rounds t after which the OLS estimate $\hat{\theta}_t$ is precise enough to ensure the identification of the best edge-arm, with high probability.

Let $\delta \in (0, 1)$ and let $\mathbf{A}_t = \sum_{i=1}^{mt} z_i z_i^\top$ be the matrix computed with the mt edge-arms constructed during t rounds. Following the steps of Soare et al. (2014), we can show that if there exists $z \in \mathcal{Z}$ such that for all $z' \in \mathcal{Z}$ the following holds:

$$\|z - z'\|_{\mathbf{A}_t^{-1}} \sqrt{8\sigma^2 \log\left(\frac{6m^2t^2K^4}{\delta\pi^2}\right)} \leq \hat{\Delta}_t(z, z') \ , \quad (3)$$

then with probability at least $1 - \delta$, the OLS estimate $\hat{\theta}_t$ leads to the best edge-arm. Details of the derivation are given in Appendix B.

As mentioned in Soare et al. (2014), by noticing that $\max_{(z,z')\in\mathcal{Z}^2} \|z - z'\|_{\mathbf{A}_t^{-1}} \leq 2\max_{z\in\mathcal{Z}} \|z\|_{\mathbf{A}_t^{-1}}$, an admissible strategy is to pull edge-arms minimizing $\max_{z\in\mathcal{Z}} \|z\|_{\mathbf{A}_t^{-1}}$ in order to satisfy the stopping condition as soon as possible. More formally, one wants to find the sequence of edge-arms $\mathbf{z}^\star_{mt} = (z^\star_1, \dots, z^\star_{mt})$ such that:

$$\mathbf{z}^\star_{mt} \in \underset{(z_1,\dots,z_{mt})}{\arg\min} \max_{z' \in \mathcal{Z}} z'^\top \left(\sum_{i=1}^{mt} z_i z_i^\top \right)^{-1} z' \ . \tag{G-opt-$\mathcal{Z}$}$$

This is known as *G-allocation* (see *e.g.,* Pukelsheim (2006); Soare et al. (2014)) and is NP-hard to compute (Çivril and Magdon-Ismail, 2009; Welch, 1982). One way to find an approximate solution is to rely on a convex relaxation of the optimization problem (G-opt-$\mathcal{Z}$) and first compute a real-valued allocation $\lambda^\star \in \mathcal{S}_\mathcal{Z}$ such that

$$\lambda^\star \in \underset{\lambda \in \mathcal{S}_\mathcal{Z}}{\arg\min} \max_{z' \in \mathcal{Z}} z'^\top \left(\sum_{z \in \mathcal{Z}} \lambda_z z z^\top \right)^{-1} z' \ . \tag{G-relaxed-$\mathcal{Z}$}$$

One could either use random sampling to draw edge-arms as i.i.d. samples from the $\lambda^\star$ distribution or rounding procedures to efficiently convert each $\lambda^\star_z$ into an integer. However, these methods do not take into account the graphical structure of the problem, and at a given round, the m chosen edge-arms may result in two different assignments for the same node. Therefore, random sampling or rounding procedures cannot be straightforwardly used to select edge-arms in $\mathcal{Z}$. Nevertheless, (G-relaxed-$\mathcal{Z}$) still gives a valuable information on the number of times, in proportion, each edge-arm $z \in \mathcal{Z}$ must be allocated to the graph. In the next section, we present an algorithm satisfying both the proportion requirements and the graphical constraints.

5.2. Random Allocation over the Nodes

Our algorithm is based on a randomized method directly allocating node-arms to the nodes and thus avoiding the difficult task of choosing edge-arms and trying to allocate them to the graph while ensuring that every node has an unique assignment. The validity of this random allocation is based on Theorem 5.1 below showing that one can draw node-arms in $\mathcal{X}$ and allocate them to the graph such that the associated edge-arms follow the probability distribution $\lambda^\star$ solution of (G-relaxed-$\mathcal{Z}$).

Theorem 5.1. *Let $\mu^\star$ be a solution of the following optimization problem:*

$$\min_{\mu \in \mathcal{S}_\mathcal{X}} \max_{x' \in \mathcal{X}} x'^\top \left(\sum_{x \in \mathcal{X}} \mu_x x x^\top \right)^{-1} x' \ . \tag{G-relaxed-$\mathcal{X}$}$$

Let $\lambda^\star \in \mathcal{S}_\mathcal{Z}$ be defined for all $z = \mathrm{vec}\left(xx'^\top\right) \in \mathcal{Z}$ by $\lambda^\star_z = \mu^\star_x \mu^\star_{x'}$. Then, $\lambda^\star$ is a solution of (G-relaxed-$\mathcal{Z}$).

Sketch of proof. The objective at the optimum in (G-relaxed-$\mathcal{X}$) and (G-relaxed-$\mathcal{Z}$) are respectively equal to d and d^2 which is the dimension of their respective problem, a result known as the Equivalence Theorem (Kiefer and Wolfowitz, 1960). Thus, by multiplying the optimum value of (G-relaxed-$\mathcal{X}$) by itself, we can show that for all $z \in \mathcal{Z}$ where $z = \mathrm{vec}\left(xx'^\top\right)$ with $(x, x') \in \mathcal{X}^2$, $\lambda^\star_z$ can be written as the product $\mu^\star_x \mu^\star_{x'}$. We refer to the Appendix C for the detailed proof.

This theorem implies that, at each round $t > 0$ and each node $i \in V$, if $x_t^{(i)}$ is drawn from $\mu^\star$, then for all pairs of neighbors $(i, j) \in E$ the probability distribution of the associated edge-arms $z_t^{(i,j)}$ follows $\lambda^\star$. Moreover, as $\mu^\star$ is a distribution over the node-arm set $\mathcal{X}$, $\lambda^\star$ is a joint (product) probability distribution on $\mathcal{X}^2$ with marginal $\mu^\star$.

We apply the Frank-Wolfe algorithm (Frank et al., 1956) to compute the solution $\mu^\star$ of (G-relaxed-$\mathcal{X}$), as it is more suited to optimization tasks on the simplex than projected gradient descent. Although we face a min-max optimization problem, we notice that the function $h(\mu) = \max_{x' \in \mathcal{X}} x'^\top \left(\sum_{x \in \mathcal{X}} \mu_x x x^\top\right)^{-1} x'$ is convex. We refer the reader to Appendix F and references therein for a proof on the convexity of h and a discussion about using Frank-Wolfe for solving (G-relaxed-$\mathcal{X}$).

Given the characterization in Theorem 5.1 and our objective to verify the stopping condition in (3), we present our sampling procedure in Algorithm 2. We also note that at each round the sampling of the node-arms can be done in parallel.

Algorithm 2 Randomized G-Allocation strategy for Graphical Bilinear Bandits

```
Input   : graph $\mathcal{G} = (V, E)$, arm set $\mathcal{X}$
Set $A_0 = I$ ; $b_0 = 0$ ; $t = 1$;
Apply the Frank-Wolfe algorithm to find $\mu^\star$ solution
  of (G-relaxed-$\mathcal{X}$).
while stopping condition (3) is not verified do
    // Sampling the node-arms
    Draw $x_t^{(1)}, \dots, x_t^{(n)} \overset{\text{iid}}{\sim} \mu^\star$ and obtain for all $(i, j)$ in $E$
      the rewards $r_t^{(i,j)}$;
    // Estimating $\hat{\theta}_t$ with the associated edge-arms
    $\mathbf{A}_t = \mathbf{A}_{t-1} + \sum_{(i,j) \in E} z_t^{(i,j)} z_t^{(i,j)\top}$;
    $b_t = b_{t-1} + \sum_{(i,j) \in E} z_t^{(i,j)} r_t^{(i,j)}$;
    $\hat{\theta}_t = \mathbf{A}_t^{-1} b_t$
    $t \leftarrow t + 1$;
end
return $\hat{\theta}_t$
```

This sampling procedure implies that each edge-arm follows the optimal distribution $\lambda^\star$. However, if we take the number of times each $z \in \mathcal{Z}$ appears in the m pulled edge-arms of a given round, we might notice that the observed proportion is not close to $\lambda_z^\star$, regardless of the size of m. This is due to the fact that the m edge-arms are not independent because of the graph structure (*cf.* Section 6). Conversely, since each group of m edge-arms are independent from one round to another, the proportion of each $z \in \mathcal{Z}$ observed among the mt pulled edge-arms throughout t rounds is close to $\lambda_z^\star$.

One may wonder if deterministic rounding procedures could be used instead of random sampling on $\mu^\star$, as it is done in many standard linear bandit algorithms (Soare et al., 2014; Fiez et al., 2019). Applying rounding procedure on $\mu^\star$ gives the number of times each node-arm $x \in \mathcal{X}$ should be allocated to the graph. However, it does not provide the actual allocations that the learner must choose over the t rounds to optimally pull the associated edge-arms (*i.e.,* pull edge-arms following $\lambda^\star$). Thus, although rounding procedures give a more precise number of times each node-arm should be pulled, the problem of allocating them to the graph remains open, whereas by concentration of the measure, randomized sampling methods imply that the associated edge-arms follow the optimal probability distribution $\lambda^\star$. In this paper, we present a simple and standard randomized G-allocation strategy, but other more elaborated methods could be considered, as long as they include the necessary randomness.

On the choice of the G-allocation problem. We have considered the G-allocation optimization problem (G-opt-$\mathcal{Z}$), however, one could want to directly minimize $\max_{(z,z')\in\mathcal{Z}^2} \|z - z'\|_{\mathbf{A}_t^{-1}}$, known as the XY-allocation (Soare et al., 2014; Fiez et al., 2019). Hence, one may want to construct edge-arms that follow the distribution $\lambda^\star_{\mathrm{XY}}$ solution of the relaxed XY-allocation problem:

$$\min_{\lambda} \max_{z',z''} \ (z'-z'')^\top \left(\sum_{z\in\mathcal{Z}} \lambda_z z z^\top \right)^{-1} (z'-z'') \ \ .$$

Although efficient in the linear case, this approach outputs a distribution $\lambda^\star_{\mathrm{XY}}$ which is not a joint probability distribution of two independent random variables, and so cannot be decomposed as the product of its marginals. Hence, there is no algorithm that allocates *identically* and *independently* the nodes of the graph to create edge-arms following $\lambda^\star_{\mathrm{XY}}$. Thus, we will rather deal with the upper bound given by the G-allocation as it allows sampling over the nodes.

Static design versus adaptive design. Adaptive designs as proposed for example in Soare et al. (2014) and Fiez et al. (2019) provide a strong improvement over static designs in the case of linear bandits. In our particular setting however, it is crucial to be able to adapt the edge-arms sampling rule to the node-arms, which is possible thanks to Theorem 5.1. This result requires a set of edge-arms $\mathcal{Z}$ expressed as a product of node-arms set $\mathcal{X}$. Extending the adaptive design of Fiez et al. (2019) to our setting would eliminate edge-arms from $\mathcal{Z}$ at each phase, without trivial guarantees that the newly obtained edge-arms set $\mathcal{Z}' \subset \mathcal{Z}$ could still be derived from another node-arms set $\mathcal{X}' \subset \mathcal{X}$. An adaptive approach is definitely a natural and promising extension of our method, and is left for future work.

5.3. Convergence Analysis

We now prove the validity of the random sampling procedure detailed in Algorithm 2 by controlling the quality of the approximation $\max_{z\in\mathcal{Z}} z^\top \mathbf{A}_t^{-1} z$ with respect to the optimum of the G-allocation optimization problem $\max_{z'\in\mathcal{Z}} z'^\top \left(\sum_{i=1}^{mt} z_i^\star z_i^{\star\top}\right)^{-1} z'$ described in (G-opt-$\mathcal{Z}$). As is usually done in the optimal design literature (see *e.g.,* Pukelsheim (2006); Soare et al. (2014); Sagnol (2010)) we bound the relative error α:

$$\max_{z\in\mathcal{Z}} z^\top \mathbf{A}_t^{-1} z \le (1+\alpha) \max_{z'\in\mathcal{Z}} z'^\top \left(\sum_{i=1}^{mt} z_i^\star z_i^{\star\top} \right)^{-1} z' \ \ .$$

Our analysis relies on several results from matrix concentration theory. One may refer for instance to Tropp et al. (2015) and references therein for an extended introduction on that matter. We first introduce a few additional notations.

Let $f_{\mathcal{Z}}$ be the function such that for any non-singular matrix $\mathbf{Q} \in \mathbb{R}^{d^2\times d^2}$, $f_{\mathcal{Z}}(\mathbf{Q}) = \max_{z\in\mathcal{Z}} z^\top \mathbf{Q}^{-1} z$ and for any distribution $\lambda \in \mathcal{S}_{\mathcal{Z}}$ let $\Sigma_{\mathcal{Z}}(\lambda) \triangleq \sum_{z\in\mathcal{Z}} \lambda_z z z^\top$ be the associated covariance matrix. Finally let $\mathbf{A}_t^\star = \sum_{i=1}^{mt} z_i^\star z_i^{\star\top}$ be the G-optimal design matrix constructed during t rounds.

For $i \in \{1, \dots, n\}$ and $s \in \{1, \dots, t\}$, let $X_s^{(i)}$ be i.i.d. random vectors in $\mathcal{X}$ such that for all $x \in \mathcal{X}$,

$$\mathbb{P}\left(X_1^{(1)} = x \right) = \mu_x^\star \ .$$

Each $X_s^{(i)}$ is to be viewed as the random arm pulled at round s for the node i. Using this notation, the random design matrix $\mathbf{A}_t$ can be defined as

$$\mathbf{A}_t = \sum_{s=1}^{t} \sum_{(i,j)\in E} \mathrm{vec}\left(X_s^{(i)} X_s^{(j)\top} \right) \mathrm{vec}\left(X_s^{(i)} X_s^{(j)\top} \right)^\top \ .$$

One can first observe that $f_{\mathcal{Z}}(\mathbf{A}_t)$ can be bounded by the following quantity:

$$\begin{aligned} f_{\mathcal{Z}}(\mathbf{A}_t) &= \max_{z\in\mathcal{Z}} \ z^\top \left(\mathbf{A}_t^{-1} - (\mathbb{E}\mathbf{A}_t)^{-1} + (\mathbb{E}\mathbf{A}_t)^{-1} \right) z \\ &\le \max_{z\in\mathcal{Z}} \ z^\top \left(\mathbf{A}_t^{-1} - (\mathbb{E}\mathbf{A}_t)^{-1} \right) z \\ &\quad + f_{\mathcal{Z}}\left(mt\Sigma_{\mathcal{Z}}(\lambda^\star) \right) \\ &\le \max_{z\in\mathcal{Z}} \|z\|^2 \| \mathbf{A}_t^{-1} - (\mathbb{E}\mathbf{A}_t)^{-1} \| \\ &\quad + f_{\mathcal{Z}}\left(\mathbf{A}_t^\star \right). \end{aligned}$$

Hence, one needs a bound on the maximum eigenvalue of $\mathbf{A}_t^{-1} - (\mathbb{E}\mathbf{A}_t)^{-1}$. Simple linear algebra leads to:

$$\mathbf{A}_t^{-1} - (\mathbb{E}\mathbf{A}_t)^{-1} = \mathbf{A}_t^{-1} (\mathbb{E}\mathbf{A}_t - \mathbf{A}_t)(\mathbb{E}\mathbf{A}_t)^{-1}.$$

Thus, in addition to bounding the maximum eigenvalue of $\mathbf{A}_t^{-1}$, which is equal to the minimum eigenvalue of $\mathbf{A}_t$, we need a bound on $\|\mathbf{A}_t - \mathbb{E}\mathbf{A}_t\|$. It may be derived from concentration results on sum of random matrices derived in Tropp et al. (2015). We now state the result controlling the relative error obtained with our randomized sampling allocation. The proof can be found in the Appendix C.

Theorem 5.2. *Let $\lambda^\star$ be a solution of the optimization problem (G-relaxed-$\mathcal{Z}$). Let $0 \leq \delta \leq 1$ and let t_0 be such that*

$$t_0 = 2Ld^2 \log(2d^2/\delta)/\nu_{\min},$$

where $L = \max_{z\in\mathcal{Z}} \|z\|^2$ and $\nu_{\min}$ is the smallest eigenvalue of the covariance matrix $\frac{1}{K^2}\sum_{z\in\mathcal{Z}} zz^\top$. Then, at each round $t \geq t_0$ with probability at least $1-\delta$, the randomized G-allocation strategy for graphical bilinear bandit in Algorithm 2 produces a matrix $\mathbf{A}_t$ such that:

$$f_{\mathcal{Z}}(\mathbf{A}_t) \leq (1+\alpha) f_{\mathcal{Z}}(\mathbf{A}_t^\star)$$

where

$$\alpha = \frac{Ld^2}{m\nu_{\min}^2}\sqrt{\frac{2v}{t}\log\left(\frac{2d^2}{\delta}\right)} + o\left(\frac{1}{\sqrt{t}}\right),$$

and $v \triangleq \mathbb{E}[(\mathbf{A}_1 - \mathbb{E}\mathbf{A}_1)^2]$.

We have just shown that the approximation value $\max_{z\in\mathcal{Z}} z^\top \mathbf{A}_t^{-1} z$ converges to the optimal value with a rate of $O\left(\sqrt{v}/(m\sqrt{t})\right)$. In Section 6, we show that the best case graph implies a $v = O\left(m\right)$ matching the convergence rate $O\left(1/\sqrt{mt}\right)$ of a linear bandit algorithm using randomized sampling to pull mt edge-arms without (graphical) constraints. Moreover, we will see that the worst case graph implies that $v = O\left(m^2\right)$.

Since we filled the gap between our constraint objective and the problem of best arm identification in linear bandits, thanks to Theorem 5.1 and 5.2, we are able to extend known results for best arm identification in linear bandits on the sample complexity and its associated lower bound.

Corollary 5.3 (Soare et al. (2014), Theorem 1)**.** *If the G-allocation is implemented with the random strategy of Algorithm 2, resulting in an α-approximation, then with probability at least $1-\delta$, the best arm obtained with $\hat{\theta}_t$ is $z_\star$ and*

$$t \leq \frac{128\sigma^2 d^2 (1+\alpha)\log\left(\frac{6m^2t^2K^4}{\delta\pi}\right)}{m\Delta_{\min}^2},$$

where $\Delta_{\min} = \min_{z\in\mathcal{Z}\setminus\{z_\star\}}(z_\star - z)^\top\theta_\star$.

Moreover, let τ be the number of rounds sufficient for any algorithm to determine the best arm with probability at least $1-\delta$. A lower bound on the expectation of τ can be obtained from the one derived for the problem of best arm identification in linear bandits (see *e.g.,* Theorem 1 in Fiez et al. (2019)):

$$\mathbb{E}[\tau] \geq \min_{\lambda\in\mathcal{S}_{\mathcal{Z}}} \max_{z\in\mathcal{Z}\setminus\{z_\star\}} \log\left(\frac{1}{2.4\delta}\right)\frac{2\sigma^2\|z_\star - z\|^2_{\Sigma_{\mathcal{Z}}(\lambda)^{-1}}}{m\left(\left(z_\star - z\right)^\top\theta_\star\right)^2}.$$

As observed in Soare et al. (2014) this lower bound can be upper bounded, in the worst case, by $4\sigma^2 d^2/(m\Delta_{\min}^2)$ which matches our bound up to log terms and the relative error α.

6. Influence of the Graph Structure on v

The convergence bound in Theorem 5.2 depends on $v = \mathbb{E}[(\mathbf{A}_1 - \mathbb{E}\mathbf{A}_1)^2]$. In this section, we characterize the impact of the graph structure on this quantity and, by extension, on the convergence rate. First of all, recall that

$$\mathbf{A}_1 = \sum_{(i,j)\in E} \operatorname{vec}\left(X_1^{(i)} X_1^{(j)\top}\right)\operatorname{vec}\left(X_1^{(i)} X_1^{(j)\top}\right)^\top.$$

Let denote $\mathbf{A}_1^{(i,j)} = \operatorname{vec}(X_1^{(i)}X_1^{(j)\top})\operatorname{vec}(X_1^{(i)}X_1^{(j)\top})^\top$ such that $\mathbf{A}_1 = \sum_{(i,j)\in E}\mathbf{A}_1^{(i,j)}$ and let define for any random matrices $\mathbf{A}$ and $\mathbf{B}$ the operators $\operatorname{Var}(\mathbf{A}) \triangleq \mathbb{E}[(\mathbf{A} - \mathbb{E}[\mathbf{A}])^2]$ and $\operatorname{Cov}(\mathbf{A},\mathbf{B}) \triangleq \mathbb{E}[(\mathbf{A} - \mathbb{E}[\mathbf{A}])(\mathbf{B} - \mathbb{E}[\mathbf{B}])]$. We can derive the variance of $\mathbf{A}_1$ as follows:

$$\begin{aligned}\operatorname{Var}\left(\mathbf{A}_1\right) &= \sum_{(i,j)\in E}\operatorname{Var}\left(\mathbf{A}_1^{(i,j)}\right)\\ &\quad + \sum_{(i,j)\in E}\sum_{\substack{(k,l)\in E\\(k,l)\neq(i,j)}}\operatorname{Cov}(\mathbf{A}_1^{(i,j)},\mathbf{A}_1^{(k,l)}).\end{aligned}$$

One can decompose the sum of the covariances into three groups: a first group where $k \neq i,j$ and $l \neq i,j$ which means that the two edges do not share any node and $\operatorname{Cov}(\mathbf{A}_1^{(i,j)},\mathbf{A}_1^{(k,l)}) = \mathbf{0}$, and two other groups where the edges share at least one node. For all edges $(i,j)\in E$ we consider either the edges $(i,k)\in E$ where $k \neq j$, yielding $\operatorname{Cov}(\mathbf{A}_1^{(i,j)},\mathbf{A}_1^{(i,k)})$ or the edges $(j,k)\in E$, yielding $\operatorname{Cov}(\mathbf{A}_1^{(i,j)},\mathbf{A}_1^{(j,k)})$.

Hence, one has

$$\begin{aligned}\operatorname{Var}\left(\mathbf{A}_1\right) &= \sum_{(i,j)\in E}\operatorname{Var}\left(\mathbf{A}_1^{(i,j)}\right)\\ &\quad + \sum_{i=1}^{n}\sum_{j\in\mathcal{N}(i)}\sum_{\substack{k\in\mathcal{N}(i)\\k\neq j}}\operatorname{Cov}(\mathbf{A}_1^{(i,j)},\mathbf{A}_1^{(i,k)})\\ &\quad + \sum_{i=1}^{n}\sum_{j\in\mathcal{N}(i)}\sum_{k\in\mathcal{N}(j)}\operatorname{Cov}(\mathbf{A}_1^{(i,j)},\mathbf{A}_1^{(j,k)}).\end{aligned}$$

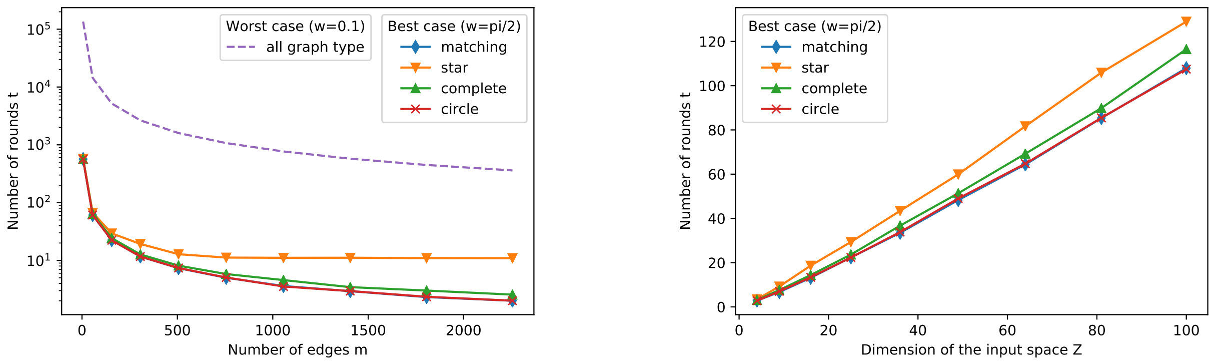

Figure 1. Number of rounds t needed to verify the stopping condition (3) with respect to **left:** the number of edges m where the dimension of the edge-arm space $\mathcal{Z}$ is fixed and equal to 25 and **right:** the dimension of the edge-arm space $\mathcal{Z}$ where the number of edges is fixed and equal to 156. For both experiments we run 100 times and plot the average number of rounds needed to verify the stopping condition.

Let $P \geq 0$ be such that for all $(i,j) \in E$, $\text{Var}\left(\mathbf{A}_1^{(i,j)}\right) \preceq P \times \mathbf{I}$ and $M, N \geq 0$ such that for all $(i,j) \in E$:

$$\forall k \in \mathcal{N}(i), \text{Cov}(\mathbf{A}_1^{(i,j)}, \mathbf{A}_1^{(i,k)}) \preceq M \times \mathbf{I}$$
$$\forall k \in \mathcal{N}(j), \text{Cov}(\mathbf{A}_1^{(i,j)}, \mathbf{A}_1^{(j,k)}) \preceq N \times \mathbf{I}$$

We want to compare the quantity $\|\text{Var}(\mathbf{A}_1)\|$ for different types of graphs: star, complete, circle and a matching graph. To have a fair comparison, we want graphs that reveal the same number of rewards at each round of the learning procedure. Hence, we denote respectively n_{S}, n_{Co}, n_{Ci} and n_{M} the number of nodes in a star, complete, circle and matching graph of m edges and get:

Star graph:

$$\|\text{Var}(\mathbf{A}_1)\| \leq mP + n_{\text{S}}^2(M+N).$$

Complete graph:

$$\|\text{Var}(\mathbf{A}_1)\| \leq mP + n_{\text{Co}}^3(M+N).$$

Circle graph:

$$\|\text{Var}(\mathbf{A}_1)\| \leq mP + n_{\text{Ci}}(2M+4N).$$

Matching graph:

$$\|\text{Var}(\mathbf{A}_1)\| \leq mP + n_{\text{M}}N.$$

We refer the reader to Appendix D for more details on the given upper bounds. Since the star (respectively, complete, circle and matching) graph of m edges has a number of nodes $n_{\text{S}} = m/2 + 1$ (respectively $n_{\text{Co}} = \left(1+\sqrt{4m+1}\right)/2$, $n_{\text{Ci}} = m/2$ and $n_{\text{M}} = m$), we obtain the bounds stated in Table 1.

Graph	Upper bound on $\|\text{Var}(\mathbf{A}_1)\|$	α
Star	$mP + (M+N)O\left(m^2\right)$	$O\left(1/\sqrt{t}\right)$
Complete	$mP + (M+N)O\left(m\sqrt{m}\right)$	$O\left(1/\left(m^{\frac{1}{4}}\sqrt{t}\right)\right)$
Circle	$mP + (M+N)O\left(m\right)$	$O\left(1/\sqrt{mt}\right)$
Matching	$mP + mN$	$O\left(1/\sqrt{mt}\right)$

Table 1. Upper bound on the variance and convergence rate of Algorithm 2 for the star, complete, circle and matching graph with respect to the number of edges m and the number of rounds t.

These four examples evidence the strong dependency of the variance on the structure of the graph. The more independent the edges are (*i.e.,* with no common nodes), the smaller the quantity $\|\text{Var}(\mathbf{A}_1)\|$ is. For a fixed number of edges m, the best case is the matching graph where no edge share the same node and the worst case is the star graph where all the edges share a central node.

7. Experiments

In this section, we consider the modified version of a standard experiment introduced by Soare et al. (2014) and used in most papers on best arm identification in linear bandits (Xu et al., 2018; Tao et al., 2018; Fiez et al., 2019; Zaki et al., 2019) to evaluate the sample complexity of our algorithm on different graphs. We consider $d+1$ node-arms in $\mathcal{X} \subset \mathbb{R}^d$ where $d \geq 2$. This node-arm set is made of the d vectors $(\mathbf{e}_1, \ldots, \mathbf{e}_d)$ forming the canonical basis of $\mathbb{R}^d$ and one additional arm $x_{d+1} = (\cos(\omega), \sin(\omega), 0, \ldots, 0)^\top$ with $\omega \in]0, \pi/2]$. Note that by construction, the edge-arm set $\mathcal{Z}$ contains the canonical basis $(\mathbf{e}'_1, \ldots, \mathbf{e}'_{d^2})$ of $\mathbb{R}^{d^2}$. The parameter matrix $\mathbf{M}_\star$ has its first coordinate equal to 2 and the others equal to 0 which makes $\theta_\star = \text{vec}(\mathbf{M}_\star) = (2, 0, \ldots, 0)^\top \in \mathbb{R}^{d^2}$. The best edge-arm is thus $z_\star = z^{(1,1)} = \mathbf{e}'_1$. One can note that when ω tends to 0, it is harder to differentiate this arm from $z^{(d+1,d+1)} = \text{vec}\left(x_{(d+1)} x_{(d+1)}^\top\right)$ than from the other arms. We set $\eta_t^{(i,j)} \sim \mathcal{N}(0, 1)$, for all edges (i, j) and round t.

We consider the two cases where $\omega = 0.1$ which makes the edge-arms $z^{(1,1)}$ and $z^{(d+1,d+1)}$ difficult to differentiate, and $\omega = \pi/2$ which makes the edge-arm $z^{(1,1)}$ easily identifiable as the optimal edge-arm. For each of these two cases, we evaluate the influence of the graph structure, the number of edges m and the edge-arm space dimension d^2 on the sampling complexity. Results are shown in Figure 1.

When $\omega = 0.1$, the type of the graph does not impact the number of rounds needed to verify the stopping condition. This is mainly due to the fact that the magnitude of its associated variance is negligible with respect to the number of rounds. Hence, even if we vary the number of edges or the dimension, we get the same performance for any type of graph including the matching graph. This implies that our algorithm performs as well as a linear bandit that draws m edge-arms in parallel at each round. When $\omega = \pi/2$, the number of rounds needed to verify the stopping condition is smaller and the magnitude of the variance is no longer negligible. Indeed, when the number of edges or the dimension increases, we notice that the star graph takes more times to satisfy the stopping condition. Moreover, note that the sample complexities obtained for the circle and the matching graph are similar. This observation is in line with the dependency on the variance shown in Table 1.

8. Conclusion

We introduced a new graphical bilinear bandit setting and studied the best arm identification problem with a fixed confidence. This problem being NP-Hard even with the knowledge of the true parameter matrix $\mathbf{M}^\star$, we first proposed an algorithm that provides a 1/2-approximation. Then, we provided a second algorithm, based on G-allocation strategy, that uses randomized sampling over the nodes to return a good estimate $\hat{\mathbf{M}}$ that can be used instead of $\mathbf{M}_\star$. Finally, we highlighted the impact of the graph structure on the convergence rate of our algorithm and validated our theoretical results with experiments. Promising extensions of the model include considering unknown parameters $\mathbf{M}_\star^{(i,j)}$, different for each edge (i, j) of the graph, and investigating XY-allocation strategies.

Acknowledgements

We thank anonymous reviewers, whose comments helped us improve the paper significantly.

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320.

Amin, K., Kearns, M., and Syed, U. (2011). Graphical models for bandit problems. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, page 1–10.

Audibert, J.-Y. and Bubeck, S. (2010). Best arm identification in multi-armed bandits. In *Proceedings of the 23th Annual Conference on Learning Theory*, pages 41–53.

Audibert, J.-Y., Bubeck, S., and Lugosi, G. (2011). Minimax policies for combinatorial prediction games. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 107–132.

Bargiacchi, E., Verstraeten, T., Roijers, D., Nowé, A., and Hasselt, H. (2018). Learning to coordinate with coordination graphs in repeated single-stage multi-agent decision problems. In *International conference on machine learning*, pages 482–490.

Bubeck, S., Munos, R., and Stoltz, G. (2009). Pure exploration in multi-armed bandits problems. In *International conference on Algorithmic learning theory*, pages 23–37. Springer.

Cao, T. and Krishnamurthy, A. (2019). Disagreement-based combinatorial pure exploration: Sample complexity bounds and an efficient algorithm. In *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99, pages 558–588.

Cesa-Bianchi, N., Gentile, C., and Zappella, G. (2013). A gang of bandits. In *Advances in Neural Information Processing Systems*, pages 737–745.

Cesa-Bianchi, N. and Lugosi, G. (2012). Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404 – 1422.

Chen, S., Lin, T., King, I., Lyu, M. R., and Chen, W. (2014). Combinatorial pure exploration of multi-armed bandits. In *Advances in Neural Information Processing Systems*, volume 27, pages 379–387.

Chen, W., Wang, Y., and Yuan, Y. (2013). Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*, pages 151–159.

Damla Ahipasaoglu, S., Sun, P., and Todd, M. J. (2008). Linear convergence of a modified frank–wolfe algorithm for computing minimum-volume enclosing ellipsoids. *Optimisation Methods and Software*, 23(1):5–19.

Degenne, R., Ménard, P., Shang, X., and Valko, M. (2020). Gamification of pure exploration for linear bandits. *arXiv preprint arXiv:2007.00953*.

Du, Y., Kuroki, Y., and Chen, W. (2020). Combinatorial pure exploration with full-bandit or partial linear feedback. *arXiv e-prints*, pages arXiv–2006.

Erdos, P. (1975). Problems and results on finite and infinite graphs. In *Recent advances in graph theory (Proc. Second Czechoslovak Sympos., Prague, 1974)*, pages 183–192.

Fiez, T., Jain, L., Jamieson, K. G., and Ratliff, L. (2019). Sequential experimental design for transductive linear bandits. In *Advances in Neural Information Processing Systems*, pages 10667–10677.

Frank, M., Wolfe, P., et al. (1956). An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110.

Guestrin, C., Lagoudakis, M. G., and Parr, R. (2002). Coordinated reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, page 227–234.

Hassin, R. and Khuller, S. (2001). z-approximations. *Journal of Algorithms*, 41(2):429–442.

Jedra, Y. and Proutiere, A. (2020). Optimal best-arm identification in linear bandits. *arXiv preprint arXiv:2006.16073*.

Jourdan, M., Mutý, M., Kirschner, J., and Krause, A. (2021). Efficient pure exploration for combinatorial bandits with semi-bandit feedback. In *Proceedings of the 31st International Conference on Algorithmic Learning Theory*.

Jun, K.-S., Willett, R., Wright, S., and Nowak, R. (2019). Bilinear bandits with low-rank structure. In *International Conference on Machine Learning*, pages 3163–3172.

Kazerouni, A. and Wein, L. M. (2019). Best arm identification in generalized linear bandits. *arXiv preprint arXiv:1905.08224*.

Kiefer, J. and Wolfowitz, J. (1960). The equivalence of two extremum problems. *Canadian Journal of Mathematics*, 12:363–366.

Mannor, S. and Shamir, O. (2011). From bandits to experts: On the value of side-observations. In *Advances in Neural Information Processing Systems*, pages 684–692.

Perrault, P., Boursier, E., Valko, M., and Perchet, V. (2020). Statistical efficiency of thompson sampling for combinatorial semi-bandits. In *Advances in Neural Information Processing Systems*.

Petersen, K. B. and Pedersen, M. S. (2012). The matrix cookbook, nov 2012. *URL http://www2. imm. dtu. dk/pubdb/p. php*, 3274:14.

Pukelsheim, F. (2006). *Optimal Design of Experiments*. Society for Industrial and Applied Mathematics.

Rizk, G., Colin, I., Thomas, A., and Draief, M. (2019). Refined bounds for randomized experimental design. *NeurIPS Workshop on Machine Learning with Guarantees*.

Robbins, H. (1952). Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535.

Sagnol, G. (2010). *Optimal design of experiments with application to the inference of traffic matrices in large networks: second order cone programming and submodularity*. PhD thesis, École Nationale Supérieure des Mines de Paris.

Siomina, I., Varbrand, P., and Yuan, D. (2006). Automated optimization of service coverage and base station antenna configuration in UMTS networks. *IEEE Wireless Communications*, 13(6):16–25.

Soare, M., Lazaric, A., and Munos, R. (2014). Best-arm identification in linear bandits. In *Advances in Neural Information Processing Systems*, pages 828–836.

Tao, C., Blanco, S., and Zhou, Y. (2018). Best arm identification in linear bandits with linear dimension dependency. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 4877–4886.

Tropp, J. A. et al. (2015). An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230.

Valko, M. (2020). Bandits on graphs and structures.

Valko, M., Munos, R., Kveton, B., and Kocák, T. (2014). Spectral bandits for smooth graph functions. In Xing, E. P. and Jebara, T., editors, *International conference on machine learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 46–54.

Welch, W. (1982). Algorithmic complexity: Three np-hard problems in computational statistics. *Journal of Statistical Computation and Simulation - J STAT COMPUT SIM*, 15:17–25.

Xu, L., Honda, J., and Sugiyama, M. (2018). A fully adaptive algorithm for pure exploration in linear bandits. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84, pages 843–851.

Zaki, M., Mohan, A., and Gopalan, A. (2019). Towards optimal and efficient best arm identification in linear bandits. *arXiv preprint arXiv:1911.01695*.

Zaki, M., Mohan, A., and Gopalan, A. (2020). Explicit best arm identification in linear bandits using no-regret learners. *arXiv preprint arXiv:2006.07562*.

Çivril, A. and Magdon-Ismail, M. (2009). On selecting a maximum volume sub-matrix of a matrix and related problems. *Theoretical Computer Science*, 410(47):4801 – 4811.

A. An NP-Hard Problem

A.1. Proof of Theorem 4.1

Theorem A.1. *Consider a given matrix $\mathbf{M}_\star \in \mathbb{R}^{d\times d}$ and a finite arm set $\mathcal{X} \subset \mathbb{R}^d$. Unless P=NP, there is no polynomial time algorithm guaranteed to find the optimal solution of*

$$\max_{\left(x^{(1)},\dots,x^{(n)}\right)\in\mathcal{X}^n} \sum_{(i,j)\in E} x^{(i)\top}\mathbf{M}_\star\, x^{(j)} \ .$$

Proof. We prove the statement by reduction to the Max-Cut problem. Let $\mathcal{G} = (V, E)$ be a graph with $V = \{1,\dots,n\}$. Let $\mathcal{X} = \{e_0, e_1\}$, where $e_0 = (1,0)^\top$ and $e_1 = (0,1)^\top$. Let $\mathbf{M}_\star = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. For any joint arm assignment $\left(x^{(1)}\dots x^{(n)}\right) \in \mathcal{X}^n$, let $F \subseteq E$ be defined as $F = \left\{i : x^{(i)} = e_1\right\}$. Note that

$$\sum_{(i,j)\in E} x^{(i)\top}\mathbf{M}_\star x^{(j)} = \sum_{(i,j)\in E} \mathbf{1}\left[x^{(i)} \neq x^{(j)}\right] = 2 \times \sum_{(i,j)\in E} \mathbf{1}\left[i \in F, j \notin F\right],$$

where $\mathbf{1}[\cdot]$ is the indicator function. The assignement $\left(x^{(1)},\dots,x^{(n)}\right)$ induces a cut $(F, V\backslash F)$, and the value of the assignment is *precisely* twice the value of the cut. Thus, if there was a polynomial time algorithm solving our problem, this algorithm would also solve the Max-Cut problem. □

A.2. Proof of Theorem 4.2

Theorem A.2. *Let us consider the graph $\mathcal{G} = (V, E)$, a finite arm set $\mathcal{X} \subset \mathbb{R}^d$ and the matrix $\mathbf{M}_\star$ given as input to Algorithm 1. Then, the expected global reward $r = \sum_{(i,j)\in E} x^{(i)\top}\mathbf{M}_\star x^{(j)}$ associated to the returned allocation $\mathbf{x} = \left(x^{(1)},\dots,x^{(n)}\right) \in \mathcal{X}^n$ verifies:*

$$\frac{r - r_{\min}}{r_\star - r_{\min}} \geq \frac{1}{2} \ .$$

where $r_\star$ and $r_{\min}$ are respectively the highest and lowest global reward one can obtain with the appropriate joint arm. Finally, the complexity of the algorithm is in $\mathcal{O}(K^2 + n)$.

Proof. Given the matrix $\mathbf{M}_\star$, the algorithm obtains the two node-arms $(x_\star, x'_\star) \in \mathcal{X}$ solution of

$$\max_{(x,x')\in\mathcal{X}} x^\top \mathbf{M}_\star x' \ .$$

Note that it is equivalent to obtain $z_\star$ solution of

$$\max_{z\in\mathcal{Z}} z^\top \operatorname{vec}(\mathbf{M}_\star) \ .$$

Let us analyze a round of Algorithm 1 where we assign the arm of a node in V. For sake of simplicity, we assume that node i is assigned at round i. At round i, we count the number $n_1^{(i)}$ of neighbors of i that have already been assigned the arm $x_\star$ and we count the number $n_2^{(i)}$ of neighbors of i that have already been assigned the arm $x'_\star$. Then, node i is assigned the arm least represented among its neighbors, that is arm $x_\star$ if $n_2^{(i)} \geq n_1^{(i)}$ and $x'_\star$ otherwise. Eventually, the optimal edge-arm $z_\star$ has been assigned $\max(n_1^{(i)}, n_2^{(i)})$ times among node i's neighborhood. Hence, for each node i, if we denote r_i the sum of all the rewards obtained with the edge-arms constructed only during the round i, we have

$$\begin{aligned} r_i &= \max\left(n_1^{(i)}, n_2^{(i)}\right) z_\star^\top \theta_\star + \min\left(n_1^{(i)}, n_2^{(i)}\right) z^\top \theta_\star \\ &\geq \frac{n_1^{(i)} + n_2^{(i)}}{2} (z_\star^\top \theta_\star + z^\top \theta_\star) \ . \end{aligned}$$

One can notice that the arm z can only be equal to $\operatorname{vec}\left(x_\star x_\star^\top\right)$ or $\operatorname{vec}\left(x_\star'^\top x_\star'^\top\right)$. Let assume that $\operatorname{vec}\left(x_\star x_\star^\top\right)^\top \theta_\star \leq \operatorname{vec}\left(x'_\star x_\star'^\top\right)^\top \theta_\star$ without loss of generality and let consider the worst case where z is always equal to $\operatorname{vec}\left(x_\star x_\star^\top\right)$. Since z is constructed with the same node-arm $x_\star$, the allocation that constructs at each edge the edge-arm z exists (which is allocating $x_\star$ to all the nodes), thus $m \times z^\top \theta_\star \geq r_{\min}$.

Moreover one can also notice that $m \times z_\star^\top \theta_\star \geq r_\star$. We thus have,

$$r_i \geq \frac{n_1^{(i)} + n_2^{(i)}}{2m}(r_\star + r_{\min}) \enspace .$$

Now let us sum all the rewards obtained with the constructed edge-arms at each round of the algorithm, that is the global reward r of the graph allocation returned by the proposed algorithm:

$$\begin{aligned} r &= \sum_{i=1}^{n} r_i \\ &\geq \sum_{i=1}^{n} \frac{n_1^{(i)} + n_2^{(i)}}{2m}(r_\star + r_{\min}) \\ &= \frac{1}{2}(r_\star + r_{\min}) \\ &= \frac{1}{2}(r_\star - r_{\min}) + r_{\min} \enspace . \end{aligned}$$

Moreover, the algorithm does K^2 estimation to find the best couple $(x_\star, x'_\star) \in \mathcal{X}^2$, and each of the n rounds of the algorithm is in $O(1)$. Hence the complexity is equal to $O(K^2 + n)$. □

B. Deriving the stopping condition

In this section, we remind key results to derive the stopping condition. We refer the reader to Soare et al. (2014) and references therein for additional details. Let $\mathcal{Z} \subset \mathbb{R}^{d^2}$ be the set of edge-arms and let $K^2 = |\mathcal{Z}|$. For $m, t > 0$, we consider a sequence of edge-arms $\mathbf{z}_t = (z_1, \dots, z_{mt}) \in \mathcal{Z}^{mt}$ and the corresponding noisy rewards $(r_1, \dots, r_{mt})$. We assume that the noise terms in the rewards are i.i.d., following a σ-sub-Gaussian distribution. Let $\hat{\theta}_t = \mathbf{A}_t^{-1} b_t \in \mathbb{R}^{d^2}$ be the solution of the ordinary least squares problem with $\mathbf{A}_t = \sum_{i=1}^{mt} z_i z_i^\top \in \mathbb{R}^{d^2 \times d^2}$ and $b_t = \sum_{i=1}^{k} z_i r_i \in \mathbb{R}^{d^2}$. We first recall the following property.

Proposition B.1 (Proposition 1 in Soare et al. (2014))**.** *Let $c = 2\sigma\sqrt{2}$. For every fixed sequence $\mathbf{z}_t$, with probability $1 - \delta$, for all $t > 0$ and for all $z \in \mathcal{Z}$, we have*

$$\left| z^\top \theta_\star - z^\top \hat{\theta}_t \right| \leq c \|z\|_{\mathbf{A}_t^{-1}} \sqrt{\log\left(\frac{6m^2t^2K^2}{\delta\pi}\right)} \enspace .$$

Our goal is to find the arm $z_\star$ that has the optimal expected reward $z_\star^\top \theta_\star$. In other words, we want to find an arm $z \in \mathcal{Z}$, such that for all $z' \in \mathcal{Z}$, $(z - z')^\top \theta_\star \geq 0$. However, one does not have access to $\theta_\star$, so we have to use its empirical estimate.

Let us consider a confidence set $\hat{S}(\mathbf{z}_t)$ centered at $\hat{\theta}_t \in \hat{S}(\mathbf{z}_t)$ and such that $\mathbb{P}\left(\theta_\star \notin \hat{S}(\mathbf{z}_t)\right) \leq \delta$, for some $\delta > 0$. Since $\theta_\star$ belongs to $\hat{S}(\mathbf{z}_t)$ with probability at least $1 - \delta$, one can stop pulling arms when an arm has been found, such that the above condition is verified for any $\theta \in \hat{S}(\mathbf{z}_t)$. More formally, the best arm identification task will be considered successful when an arm $z \in \mathcal{Z}$ will verify the following condition for any $z' \in \mathcal{Z}$ and any $\theta \in \hat{S}(\mathbf{z}_t)$:

$$(z - z')^\top (\hat{\theta}_t - \theta) \leq \hat{\Delta}_t(z, z') \enspace ,$$

where $\hat{\Delta}_t(z, z') = (z - z')^\top \hat{\theta}_t$ is the empirical gap between z and z'.

Using the upper bound in Proposition B.1, one way to ensure that $\mathbb{P}\left(\theta_\star \in \hat{S}(\mathbf{z}_t)\right) \geq 1 - \delta$ is to define the confidence set $\hat{S}(\mathbf{z}_t)$ as follows

$$\hat{S}(\mathbf{z}_t) = \left\{ \theta \in \mathbb{R}^d,\ \forall z \in \mathcal{Z},\ \forall z' \in \mathcal{Z}, (z - z')^\top \left(\hat{\theta}_t - \theta\right) \leq c\|z - z'\|_{(\mathbf{A}_t)^{-1}} \sqrt{\log\left(\frac{6m^2t^2K^4}{\delta\pi}\right)} \right\} \enspace .$$

Then, the stopping condition can be reformulated as follows:

$$\exists z \in \mathcal{Z},\ \forall z' \in \mathcal{Z},\ c\|z - z'\|_{\mathbf{A}_t^{-1}} \sqrt{\log\left(\frac{6m^2t^2K^4}{\delta\pi}\right)} \leq \hat{\Delta}_t\left(z, z'\right) \quad . \tag{4}$$

C. Estimation of the unknown parameter

C.1. Proof of Theorem 5.1

To prove Theorem 5.1, we first state some useful propositions and lemmas. For any finite set $X \subset \mathbb{R}^d$, we define the function $h_X : \mathcal{S}_X \to \mathbb{R} \cup \{+\infty\}$ as follows: for any $\lambda \in \mathcal{S}_X$,

$$h_X(\lambda) = \begin{cases} \max_{x' \in X} x'^\top \Sigma_X(\lambda)^{-1} x' & \text{if } \Sigma_X(\lambda) \text{ is invertible} \\ +\infty & \text{otherwise } . \end{cases}$$

Lemma C.1. *Let $\mathcal{X} \subset \mathbb{R}^d$ be a finite set spanning $\mathbb{R}^d$ and let $\mathcal{Z} = \{\mathrm{vec}\,(xx'^\top),\, (x, x') \in \mathcal{X}^2\}$. If $\mu^\star \in \mathcal{S}_\mathcal{X}$ is a minimizer of $h_\mathcal{X}$, then $\mu^\star$ is a solution of*

$$\min_{\mu \in \mathcal{S}_\mathcal{X}} \max_{z \in \mathcal{Z}} z^\top \left(\sum_{x \in \mathcal{X}} \sum_{x' \in \mathcal{X}} \mu_x \mu_{x'} \,\mathrm{vec}\left(xx'^\top\right) \mathrm{vec}\left(xx'^\top\right)^\top \right)^{-1} z \quad .$$

Proof. First, let us notice that, for any $\mathcal{X} \subset \mathbb{R}^d$, one has $h_\mathcal{X} \geq 0$. Thus, $\mu^\star$ is also a minimizer of $h_\mathcal{X}^2$. In addition, $\mathcal{X}$ is spanning $\mathbb{R}^d$ so $h_\mathcal{X}(\mu^\star) < +\infty$. Developing $h_\mathcal{X}(\mu^\star)^2$ yields:

$$\begin{aligned} h_\mathcal{X}(\mu^\star) \times h_\mathcal{X}(\mu^\star) &= \left(\max_{x \in \mathcal{X}}\ x^\top \Sigma_X(\mu^\star)^{-1} x \right) \times \left(\max_{x \in \mathcal{X}}\ x^\top \Sigma_X(\mu^\star)^{-1} x \right) \\ &= \max_{x \in \mathcal{X}} \max_{x' \in \mathcal{X}}\ x^\top \Sigma_X(\mu^\star)^{-1} x x'^\top \Sigma_X(\mu^\star)^{-1} x' \\ &= \max_{x \in \mathcal{X}} \max_{x' \in \mathcal{X}}\ \mathrm{vec}\left(xx'^\top\right)^\top \mathrm{vec}\left(\Sigma_X(\mu^\star)^{-1} x x'^\top \Sigma_X(\mu^\star)^{-1}\right) \\ &= \max_{x \in \mathcal{X}} \max_{x' \in \mathcal{X}}\ \mathrm{vec}\left(xx'^\top\right)^\top \left(\Sigma_X(\mu^\star)^{-1} \otimes \Sigma_X(\mu^\star)^{-1}\right) \mathrm{vec}\left(xx'^\top\right) \\ &= \max_{z \in \mathcal{Z}}\ z^\top \left(\Sigma_X(\mu^\star)^{-1} \otimes \Sigma_X(\mu^\star)^{-1}\right) z \ , \end{aligned}$$

where $\otimes$ denotes the Kronecker product. We can now focus on the central term:

$$\begin{aligned} \Sigma_X(\mu^\star)^{-1} \otimes \Sigma_X(\mu^\star)^{-1} &= \left(\sum_{x \in \mathcal{X}} \mu_x^\star x x^\top \right)^{-1} \otimes \left(\sum_{x \in \mathcal{X}} \mu_x^\star x x^\top \right)^{-1} \\ &= \left(\sum_{x \in \mathcal{X}} \mu_x^\star x x^\top \otimes \sum_{x \in \mathcal{X}} \mu_x^\star x x^\top \right)^{-1} \\ &= \left(\sum_{x \in \mathcal{X}} \sum_{x' \in \mathcal{X}} \mu_x^\star \mu_{x'}^\star \left(xx^\top \otimes x'x'^\top\right) \right)^{-1} \\ &= \left(\sum_{x \in \mathcal{X}} \sum_{x' \in \mathcal{X}} \mu_x^\star \mu_{x'}^\star \,\mathrm{vec}\left(xx'^\top\right) \mathrm{vec}\left(xx'^\top\right)^\top \right)^{-1} \quad , \end{aligned}$$

and the result holds. □

Theorem C.2. *Let $\mu^\star \in \mathcal{S}_\mathcal{X}$ be a minimizer of $h_\mathcal{X}$. Let $\lambda^\star \in \mathcal{S}_\mathcal{Z}$ be the distribution defined from $\mu^\star$ such that, for all $z = \mathrm{vec}\,(xx'^\top)$, $\lambda_z^\star = \mu_x^\star \mu_{x'}^\star$. Then $\lambda^\star$ is a minimizer of $h_\mathcal{Z}$.*

Proof. From Kiefer and Wolfowitz (1960), we know that $\min_{\lambda\in\mathcal{S}_\mathcal{Z}} h_\mathcal{Z}(\lambda) = d^2$ and $\min_{\mu\in\mathcal{S}_\mathcal{X}} h_\mathcal{X}(\mu) = d$. Then, using Proposition C.1, one has

$$\begin{aligned} d^2 &= h_\mathcal{X}(\mu^\star) \times h_\mathcal{X}(\mu^\star) \\ &= \max_{z\in\mathcal{Z}} z^\top \left(\sum_{x\in\mathcal{X}} \sum_{x'\in\mathcal{X}} \mu^\star_x \mu^\star_{x'} \operatorname{vec}\left(xx'^\top\right) \operatorname{vec}\left(xx'^\top\right)^\top \right)^{-1} z \ . \end{aligned}$$

This result implies that $h_\mathcal{Z}(\lambda^\star) = d^2$. Since $\min_{\lambda\in\mathcal{S}_\mathcal{Z}} h_\mathcal{Z}(\lambda) = d^2$, $\lambda^\star$ is a minimizer of $h_\mathcal{Z}$.

□

C.2. Proof of Theorem 5.2

To prove our confidence bound, we need the two following proposition. The first one is from (Tropp et al., 2015).

Proposition C.3 (Tropp et al. (2015), Chapter 5 and 6)**.** *Let $\mathbf{Z}_1, \dots, \mathbf{Z}_t$ be i.i.d. positive semi-definite random matrices in $\mathbb{R}^{d^2\times d^2}$, such that there exists $L > 0$ verifying $\mathbf{0} \preceq \mathbf{Z}_1 \preceq mL\mathbf{I}$. Let $\mathbf{A}_t$ be defined as $\mathbf{A}_t \triangleq \sum_{s=1}^t \mathbf{Z}_s$. Then, for any $0 < \varepsilon < 1$, one can lowerbound $\lambda_{\min}(\mathbf{A}_t)$ as follows:*

$$\mathbb{P}(\lambda_{\min}(\mathbf{A}_t) \leq (1-\varepsilon)\lambda_{\min}(\mathbb{E}\mathbf{A}_t)) \leq d^2 e^{-\dfrac{t\varepsilon^2\lambda_{\min}(\mathbb{E}\mathbf{Z}_1)}{2mL}} \ .$$

If in addition, there exists some $v > 0$, such that $\|\mathbb{E}[(\mathbf{Z}_1 - \mathbb{E}\mathbf{Z}_1)^2]\| \leq v$, then for any $u > 0$, one has

$$\mathbb{P}\left(\|\mathbf{S}_t\| \geq u\right) \leq 2d^2 e^{-\dfrac{u^2}{2mLu/3 + 2tv}},$$

From the second inequality, (Rizk et al., 2019) derived a slightly different inequality that we use here :

Proposition C.4 (Rizk et al. (2019), Appendix A.3)**.** *Let $\mathbf{Z}_1, \dots, \mathbf{Z}_t$ be t i.i.d. random symmetric matrices in $\mathbb{R}^{d^2\times d^2}$ such that there exists $L > 0$ such that $\|\mathbf{Z}_1\| \leq mL$, almost surely. Let $\mathbf{A}_t \triangleq \sum_{i=1}^t \mathbf{Z}_i$. Then, for any $u > 0$, one has:*

$$\mathbb{P}\left(\|\mathbf{A}_t - \mathbb{E}\mathbf{A}_t\| \geq \sqrt{2tvu} + \frac{mLu}{3} \right) \leq d^2 e^{-u} \ .$$

where $v \triangleq \left\|\mathbb{E}[(\mathbf{Z}_1 - \mathbb{E}\mathbf{Z}_1)^2]\right\|$.

Finally, to prove our main theorem, we need the following lemma.

Lemma C.5. *One has $\left\|\Sigma_\mathcal{Z}(\lambda^\star)^{-1}\right\| \leq \frac{d^2}{\nu_{\min}}$, where $\nu_{\min}$ is the smallest eigenvalue of the covariance matrix $\frac{1}{K^2}\sum_{z\in\mathcal{Z}} z^\top z$.*

Proof. Define $\mathcal{B} = \left\{ z \in \mathbb{R}^{d^2} : \|z\| = 1 \right\}$. First, for any semi-definite matrix $\mathbf{A} \in \mathbb{R}^{d^2\times d^2}$, we have $\|\mathbf{A}\| = \max_{z\in\mathcal{B}} \ z^\top \mathbf{A} z$. Because $\Sigma_\mathcal{Z}(\lambda^\star)^{-1}$ is positive definite and symmetric, and by Rayleigh-Ritz theorem,

$$\left\|\Sigma_\mathcal{Z}(\lambda^\star)^{-1}\right\| = \max_{z\in\mathcal{B}} \frac{z^\top \Sigma_\mathcal{Z}(\lambda^\star)^{-1} z}{z^\top z} = \max_{z\in\mathcal{B}} \ z^\top \Sigma_\mathcal{Z}(\lambda^\star)^{-1} z \ .$$

Let $\mathbf{Z} \in \mathbb{R}^{K^2\times d^2}$ be the matrix whose rows are vectors of $\mathcal{Z}$ in an arbitrary order. Notice that $\mathcal{Z}$ spans $\mathbb{R}^{d^2}$, since $\mathcal{X}$ spans $\mathbb{R}^d$. Now for any $z \in \mathcal{B}$, define $\beta^{(z)} \in \mathbb{R}^{K^2}$ as a vector such that $z = \mathbf{Z}^\top \beta^{(z)}$. Then,

$$\begin{aligned} \left\|\Sigma_\mathcal{Z}(\lambda^\star)^{-1}\right\| &= \max_{z\in\mathcal{B}} \ {\beta^{(z)}}^\top \mathbf{Z} \Sigma_\mathcal{Z}(\lambda^\star)^{-1} \mathbf{Z}^\top \beta^{(z)} \\ &= \max_{z\in\mathcal{B}} \sum_{i=1}^{d^2} \sum_{j=1}^{d^2} \beta^{(z)}_i \beta^{(z)}_j z_i^\top \Sigma_\mathcal{Z}(\lambda^\star)^{-1} z_j \\ &\leq \max_{z\in\mathcal{B}} \ \left\|\beta^{(z)}\right\|_1^2 \times \max_{i,j} \ z_i^\top \Sigma_\mathcal{Z}(\lambda^\star)^{-1} z_j \ . \end{aligned}$$

Define $\tilde{z}_i = \Sigma_{\mathcal{Z}}(\lambda^\star)^{-\frac{1}{2}} z_i$. Clearly, $\max_{i,j}\ z_i^\top \Sigma_{\mathcal{Z}}(\lambda^\star)^{-1} z_j = \max_{i,j}\ \tilde{z}_i^\top \tilde{z}_j = \max_i\ \tilde{z}_i^2$. So we have

$$\begin{aligned}\left\|\Sigma_{\mathcal{Z}}(\lambda^\star)^{-1}\right\| &\leq \max_{z\in\mathcal{B}} \left\|\beta^{(z)}\right\|_1^2 \times \max_{z'\in\mathcal{Z}} z'^\top \Sigma_{\mathcal{Z}}(\lambda^\star)^{-1} z' \\ &\leq \max_{z\in\mathcal{B}} \left\|\beta^{(z)}\right\|_1^2 d^2 \ .\end{aligned}$$

The last inequality comes from Kiefer and Wolfowitz equivalence theorem (Kiefer and Wolfowitz, 1960). Now observe that $\beta^{(z)}$ can be obtained by least square regression : $\beta^{(z)} = \left(\mathbf{Z}\mathbf{Z}^\top\right)^{-1}\mathbf{Z}z = \left(\mathbf{Z}^\top\right)^\dagger z$ where $(\cdot)^\dagger$ is the Moore-Penrose pseudo-inverse. Note that $\mathbf{Z}\mathbf{Z}^\top$ is a Gram matrix. It is known that for a matrix having singular values $\{\sigma_i\}_i$, its pseudo-inverse has singular values $\begin{cases}\frac{1}{\sigma_i} & \text{if } \sigma_i \neq 0 \\ 0 & \text{otherwise}\end{cases}$ for all i. So for $z \in \mathcal{B}$, we have:

$$\left\|\beta^{(z)}\right\|_1^2 \leq K^2 \left\|\beta^{(z)}\right\|_2^2 \leq K^2 \left\|\left(\mathbf{Z}^\top\right)^\dagger\right\|^2 \leq \frac{K^2}{\sigma_{\min}\left(\mathbf{Z}\right)^2} \ ,$$

where $\sigma_{\min}(\cdot)$ refers to the smallest singular value. Let $\nu_{\min}(\cdot)$ refer to the smallest eigenvalue. Noting that

$$\sigma_{\min}\left(\mathbf{Z}\right)^2 = \nu_{\min}\left(\mathbf{Z}^\top\mathbf{Z}\right) = K^2 \nu_{\min}\left(\frac{1}{K^2}\sum_{z\in\mathcal{Z}} z z^\top\right) \ ,$$

yields the desired result.

□

We are now ready to state the bound on the random sampling error, relatively to the objective value $\Sigma_{\mathcal{Z}}(\lambda^\star)$ of the convex relaxation solution.

Theorem C.6. *Let $\lambda^\star \in \mathcal{S}_{\mathcal{Z}}$ be a minimizer of $h_{\mathcal{Z}}$. Let $0 \leq \delta \leq 1$ and let $t_0 > 0$ be such that*

$$t_0 = 2Ld^2 \log(2d^2/\delta)/\nu_{\min} \ ,$$

where $L = \max_{z\in\mathcal{Z}} \|z\|^2$ and $\nu_{\min}$ is the smallest eigenvalue of the covariance matrix $\frac{1}{K^2}\sum_{z\in\mathcal{Z}} z^\top z$. Then, at each round $t \geq t_0$, with probability at least $1-\delta$, the randomized G-allocation strategy for graphical bilinear bandit in Algorithm 2 produces a matrix $\mathbf{A}_t$ such that:

$$h_{\mathcal{Z}}(\mathbf{A}_t) \leq (1+\alpha) h_{\mathcal{Z}}(mt \times \Sigma_{\mathcal{Z}}(\lambda^\star)) \ ,$$

where

$$\alpha = \frac{Ld^2}{m\nu_{\min}^2}\sqrt{\frac{2v}{t}\log\left(\frac{2d^2}{\delta}\right)} + o\left(\frac{1}{\sqrt{t}}\right),$$

and $v \triangleq \|\mathbb{E}[(\mathbf{A}_1 - \mathbb{E}\mathbf{A}_1)^2]\|$.

Proof. Let $(X_s^{(1)})_{s=1,\ldots,t}, \ldots, (X_s^{(n)})_{s=1,\ldots,t}$ be nt i.i.d. random vectors in $\mathbb{R}^d$ such that for all $x \in \mathcal{X}$, $\mathbb{P}\left(X_1^{(1)} = x\right) = \mu_x^\star$. For $(i,j) \in E$ and $1 \leq s \leq t$, we define the random matrix $\mathbf{Z}_s^{(i,j)}$ by

$$\mathbf{Z}_s^{(i,j)} = \operatorname{vec}\left(X_s^i X_s^{j\top}\right) \operatorname{vec}\left(X_s^i X_s^{j\top}\right)^\top \ .$$

Finally, let us define for all $1 \leq s \leq t$, the edge-wise sum $\mathbf{Z}_s \in \mathbb{R}^{d^2\times d^2}$, that is

$$\mathbf{Z}_s = \sum_{(i,j)\in E} \mathbf{Z}_s^{(i,j)} \ .$$

One can easily notice that $\mathbf{Z}_1, \dots, \mathbf{Z}_t$ are i.i.d. random matrices. We define the overall sum $\mathbf{A}_t = \sum_{s=1}^t \mathbf{Z}_s$ and our goal is to measure how close $f_{\mathcal{Z}}(\mathbf{A}_t)$ is to $f_{\mathcal{Z}}(mt \times \Sigma_{\mathcal{Z}}(\lambda^\star))$, where mt corresponds to the total number of sampled arms $z \in \mathcal{Z}$ during the t rounds of the learning procedure. By definition of $\mathbf{A}_t$, one has

$$\begin{aligned}
\max_{z \in \mathcal{Z}} z^\top \left(\mathbb{E}\mathbf{A}_t\right)^{-1} z &= \max_{z \in \mathcal{Z}} z^\top \left(\sum_{s=1}^t \sum_{(i,j)\in E} \mathbb{E}\left[\mathbf{Z}_s^{(i,j)}\right] \right)^{-1} z \\
&= \max_{z \in \mathcal{Z}} z^\top \left(\sum_{s=1}^t \sum_{(i,j)\in E} \sum_{x,x' \in \mathcal{X}} \mu_x^\star \mu_{x'}^\star \operatorname{vec}\left(xx'^\top\right) \operatorname{vec}\left(xx'^\top\right)^\top \right)^{-1} z \\
&= \max_{z \in \mathcal{Z}} z^\top \left(\sum_{s=1}^t \sum_{(i,j)\in E} \sum_{z' \in \mathcal{Z}} \lambda_{z'}^\star z' z'^\top \right)^{-1} z \\
&= f_{\mathcal{Z}}(mt\Sigma_{\mathcal{Z}}(\lambda^\star)) \ .
\end{aligned}$$

This allows us to bound the relative error as follows:

$$\begin{aligned}
\alpha &= \frac{f_{\mathcal{Z}}(\mathbf{A}_t)}{f_{\mathcal{Z}}(mt \times \Sigma_{\mathcal{Z}}(\lambda^\star))} - 1 \\
&= \frac{\max_{z\in\mathcal{Z}} \ z^\top \left(\mathbf{A}_t^{-1} - \left(\mathbb{E}\mathbf{A}_t\right)^{-1} + \left(\mathbb{E}\mathbf{A}_t\right)^{-1}\right) z}{f_{\mathcal{Z}}(mt \times \Sigma_{\mathcal{Z}}(\lambda^\star))} - 1 \\
&\leq \frac{\max_{z\in\mathcal{Z}} \ z^\top \left(\mathbf{A}_t^{-1} - \left(\mathbb{E}\mathbf{A}_t\right)^{-1}\right) z}{f_{\mathcal{Z}}(mt \times \Sigma_{\mathcal{Z}}(\lambda^\star))} \ .
\end{aligned}$$

Using the fact that $f_{\mathcal{Z}}(mt\Sigma_{\mathcal{Z}}(\lambda^\star)) = d^2/mt$ (Kiefer and Wolfowitz, 1960), we obtain

$$\begin{aligned}
\alpha &\leq \frac{mt}{d^2} \times \max_{z\in\mathcal{Z}} z^\top \left(\mathbf{A}_t^{-1} - \left(\mathbb{E}\mathbf{A}_t\right)^{-1}\right) z \\
&\leq \frac{mt}{d^2} \times \max_{z\in\mathcal{Z}} \|z\|^2 \|\mathbf{A}_t^{-1} - \left(\mathbb{E}\mathbf{A}_t\right)^{-1}\| \\
&\leq \frac{mtL}{d^2} \times \|\mathbf{A}_t^{-1} - \left(\mathbb{E}\mathbf{A}_t\right)^{-1}\| \ .
\end{aligned}$$

Therefore, controlling the quantity $\|\mathbf{A}_t^{-1} - \left(\mathbb{E}\mathbf{A}_t\right)^{-1}\|$ will allow us to provide an upper bound on the relative error. Notice that

$$\begin{aligned}
\|\mathbf{A}_t^{-1} - \left(\mathbb{E}\mathbf{A}_t\right)^{-1}\| &= \|\mathbf{A}_t^{-1}\left(\mathbb{E}\mathbf{A}_t - \mathbf{A}_t\right)\left(\mathbb{E}\mathbf{A}_t\right)^{-1}\| \\
&\leq \|\mathbf{A}_t^{-1}\| \, \|\mathbb{E}\mathbf{A}_t - \mathbf{A}_t\| \, \|\left(\mathbb{E}\mathbf{A}_t\right)^{-1}\| \ .
\end{aligned}$$

Using Proposition C.3, we know that for any $d^2 e^{-\frac{t\lambda_{\min}(\mathbb{E}\mathbf{Z}_1)}{mL}} < \delta_h < 1$, the following holds:

$$\|\mathbf{A}_t^{-1}\| \leq \frac{\|\left(\mathbb{E}\mathbf{A}_t\right)^{-1}\|}{1 - \sqrt{\frac{2mL}{t}\|\left(\mathbb{E}\mathbf{Z}_1\right)^{-1}\|\log(d^2/\delta_h)}} \ ,$$

with probability at least $1 - \delta_h$. Similarly, using Proposition C.4, for any $0 < \delta_b < 1$, we have

$$\|\mathbf{A}_t - \mathbb{E}\mathbf{A}_t\| \leq \frac{mL}{3}\log\frac{d^2}{\delta_b} + \sqrt{2tv^2\log\frac{d^2}{\delta_b}} \ ,$$

with probability at least $1 - \delta_b$. Combining these two results with a union bound leads to the following bound, with probability $1 - (\delta_b + \delta_h)$:

$$\left\|\mathbf{A}_t^{-1} - \left(\mathbb{E}\mathbf{A}_t\right)^{-1}\right\| \leq \left\|\left(\mathbb{E}\mathbf{A}_t\right)^{-1}\right\|^2 \frac{(mL/3)\log(d^2/\delta_b) + \sqrt{2tv\log(d^2/\delta_b)}}{1 - \sqrt{(2mL/t)\left\|\left(\mathbb{E}\mathbf{Z}_1\right)^{-1}\right\|\log(d^2/\delta_h)}} \ .$$

In order to obtain a unified bound depending on one confidence parameter $1-\delta$, one could optimize over δ_b and δ_h, subject to $\delta_b+\delta_h=\delta$. This leads to a messy result and a negligible improvement. One can use simple values $\delta_b=\delta_h=\delta/2$, so the overall bound becomes, with probability $1-\delta$:

$$\|\mathbf{A}_t^{-1}-(\mathbb{E}\mathbf{A}_t)^{-1}\|\leq \frac{1}{tm^2}\left\|\Sigma_{\mathcal{Z}}(\lambda^\star)^{-1}\right\|^2\sqrt{\frac{2v}{t}\log\left(\frac{2d^2}{\delta}\right)}\left(\frac{1+\sqrt{\frac{m^2L^2\log(2d^2/\delta)}{18vt}}}{1-\sqrt{\frac{2L\|\Sigma_{\mathcal{Z}}(\lambda^\star)^{-1}\|\log(2d^2/\delta)}{t}}}\right) \ .$$

This can finally be formulated as follows:

$$\left\|\mathbf{A}_t^{-1}-(\mathbb{E}\mathbf{A}_t)^{-1}\right\|\leq \frac{1}{tm^2}\left\|\Sigma_{\mathcal{Z}}(\lambda^\star)^{-1}\right\|^2\sqrt{\frac{2v}{t}\log\left(\frac{2d^2}{\delta}\right)}+o\left(\frac{1}{t\sqrt{t}}\right) \ .$$

Using the obtained bound on $\|\mathbf{A}_t^{-1}-\mathbb{E}(\mathbf{A}_t)^{-1}\|$ yields

$$\begin{aligned}\frac{f_{\mathcal{Z}}(\mathbf{A}_t)}{f_{\mathcal{Z}}(mt\times\Sigma_{\mathcal{Z}}(\lambda^\star))}-1&\leq\frac{mtL}{d^2}\times\left(\frac{1}{tm^2}\left\|\Sigma_{\mathcal{Z}}(\lambda^\star)^{-1}\right\|^2\sqrt{\frac{2v}{t}\log\left(\frac{2d^2}{\delta}\right)}+o\left(\frac{1}{t\sqrt{t}}\right)\right)\\&\leq\frac{L}{md^2}\left\|\Sigma_{\mathcal{Z}}(\lambda^\star)^{-1}\right\|^2\sqrt{\frac{2v}{t}\log\left(\frac{2d^2}{\delta}\right)}+o\left(\frac{1}{\sqrt{t}}\right) \ ,\end{aligned}$$

By noticing that $f_{\mathcal{Z}}(mt\times\Sigma_{\mathcal{Z}}(\lambda^\star))\leq f_{\mathcal{Z}}(\mathbf{A}_t^\star)$ and by using Lemma C.5, the result holds. □

D. Variance analysis

Star graph. The covariance matrix of the star graph can be bounded as follows:

$$\mathrm{Var}(\mathbf{A}_1)\preceq m\times P\cdot\mathbf{I}+(n_{\mathrm{S}}-1)(n_{\mathrm{S}}-2)M\cdot\mathbf{I}+n_{\mathrm{S}}(n_{\mathrm{S}}-1)N\cdot\mathbf{I} \ .$$

Since the star graph of m edges has a number of nodes $n_{\mathrm{S}}=m/2+1$, we have

$$\|\mathrm{Var}(\mathbf{A}_1)\|\leq m\times P+(M+N)\times O\left(m^2\right) \ .$$

Complete graph. As for the star graph,

$$\mathrm{Var}(\mathbf{A}_1)\preceq m\times P\cdot\mathbf{I}+n_{\mathrm{Co}}(n_{\mathrm{Co}}-1)(n_{\mathrm{Co}}-2)M\cdot\mathbf{I}+n_{\mathrm{Co}}(n_{\mathrm{Co}}-1)(n_{\mathrm{Co}}-1)N\cdot\mathbf{I} \ .$$

Since the complete graph of m edges has a number of nodes $n_{\mathrm{Co}}=\left(1+\sqrt{4m+1}\right)/2$, we have

$$\|\mathrm{Var}(\mathbf{A}_1)\|\leq m\times P+(M+N)\times O\left(m\sqrt{m}\right) \ .$$

Circle graph. Again,

$$\mathrm{Var}(\mathbf{A}_1)\preceq m\times P\cdot\mathbf{I}+2n_{\mathrm{Ci}}M\cdot\mathbf{I}+4n_{\mathrm{Ci}}N\cdot\mathbf{I} \ .$$

Since the circle graph of m edges has a number of nodes $n_{\mathrm{Ci}}=m/2$, we have

$$\|\mathrm{Var}(\mathbf{Z}_1)\|\leq m\times P+(M+N)\times O\left(m\right) \ .$$

Matching graph. Finally,

$$\mathrm{Var}(\mathbf{A}_1)\preceq m\times P\cdot\mathbf{I}+n_{\mathrm{M}}N\cdot\mathbf{I} \ .$$

Since the matching graph of m edges has a number of nodes $n_{\mathrm{M}}=m$, we have

$$\|\mathrm{Var}(\mathbf{A}_1)\|\leq m\times P+m\times N \ .$$

E. Generalization

In this section, we provide some insights into the generalization to broader reward settings.

E.1. When $\mathbf{M}_\star$ is not symmetric

Consider the same graphical bilinear bandit setting as the one explained in the paper with the only difference that $\mathbf{M}_\star$ is not symmetric. We recall here that in the graph $\mathcal{G} = (V, E)$ associated to the graphical bilinear bandit setting, $(i, j) \in E$ if and only if $(j, i) \in E$. Hence, for a given allocation $(x^{(1)}, \ldots, x^{(n)}) \in \mathcal{X}^n$, one can write the associated expected global reward as follows :

$$\begin{aligned}
\sum_{(i,j)\in E} x^{(i)\top}\mathbf{M}_\star x^{(j)} &= \sum_{i=1}^{n} \sum_{\substack{j\in\mathcal{N}(i)\\ j>i}} x^{(i)\top}\mathbf{M}_\star x^{(j)} + x^{(j)\top}\mathbf{M}_\star x^{(i)} \\
&= \sum_{i=1}^{n} \sum_{\substack{j\in\mathcal{N}(i)\\ j>i}} x^{(i)\top}\mathbf{M}_\star x^{(j)} + \left(x^{(j)\top}\mathbf{M}_\star x^{(i)}\right)^\top \\
&= \sum_{i=1}^{n} \sum_{\substack{j\in\mathcal{N}(i)\\ j>i}} x^{(i)\top}\mathbf{M}_\star x^{(j)} + x^{(i)\top}\mathbf{M}_\star^\top x^{(j)} \\
&= \sum_{i=1}^{n} \sum_{\substack{j\in\mathcal{N}(i)\\ j>i}} x^{(i)\top}\left(\mathbf{M}_\star x^{(j)} + \mathbf{M}_\star^\top x^{(j)}\right) \\
&= \sum_{i=1}^{n} \sum_{\substack{j\in\mathcal{N}(i)\\ j>i}} x^{(i)\top}\left(\mathbf{M}_\star + \mathbf{M}_\star^\top\right) x^{(j)} \ .
\end{aligned}$$

Let us denote $\bar{\mathbf{M}}_\star = \mathbf{M}_\star + \mathbf{M}_\star^\top$. One can notice that $\bar{\mathbf{M}}_\star$ is symmetric. Solving the graphical bilinear bandit with the matrix $\bar{\mathbf{M}}_\star$ is exactly what we propose throughout the main paper.

E.2. When the reward captures more information than the interactions between agents

Consider the real world problems introduced in the paper, but with the difference that instead of a reward only related to the interaction between two neighboring agents/nodes, there is an additional term that informs about the absolute quality of the arm chosen by the agent itself. More formally we consider the following reward $r_t^{(i,j)}$ for the node i:

$$r_t^{(i,j)} = x_t^{(i)\top}\mathbf{M}_\star x_t^{(j)} + x_t^{(i)\top}\beta_\star + \eta_t^{(i,j)} \ .$$

where $\beta_\star \in \mathbb{R}^d$ is a second unknown parameter that allows to capture the quality of the arm chosen by the node i independently of its neighbors.

In order to add a constant term in the reward, let us construct the set $\tilde{\mathcal{X}} \subset \mathbb{R}^{d+1}$ such that each arm $x \in \mathcal{X}$ is associated to a new arm $\tilde{x} \in \tilde{\mathcal{X}}$ defined as $\tilde{x}^\top = (x^\top, 1)$. Moreover, let us define the matrix $\tilde{\mathbf{M}}^\star \in \mathbb{R}^{(d+1)\times(d+1)}$ as follows:

$$\tilde{\mathbf{M}}_\star = \left(\begin{array}{cc} \left[\quad \mathbf{M}_\star \quad\right] & \left[\beta_\star\right] \\ \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix} & \end{array}\right) \ .$$

One can easily verify that for any edge $(i, j) \in E$ and any time step t, the reward $r_t^{(i,j)}$ can now be written as follows:

$$r_t^{(i,j)} = \tilde{x}_t^{(i)\top}\tilde{\mathbf{M}}_\star \tilde{x}_t^{(j)} + \eta_t^{(i,j)} \ ,$$

which leads to the same graphical bilinear bandit setting explained in Section 3, this time in dimension $d+1$ instead of d. Hence, all the previous results hold for this more general graphical bilinear bandit problem, provided any dependence in d is modified to $d+1$.

F. Computing $\mu^\star$

In Algorithm 2, we need to find the solution $\mu_\star$ of $\min_{\mu\in\mathcal{S}_\mathcal{X}} h_\mathcal{X}(\mu)$. In fact we need $\mu_\star$ to sample from it. We show that for any set X, the function h_X is convex and we use the Frank-Wolfe algorithm (Frank et al., 1956) to compute $\mu_\star$ and $\lambda_\star$. The convergence of the algorithm has been proven in Damla Ahipasaoglu et al. (2008). Note that one can only compute $\mu_\star$ or $\lambda_\star$ to obtain the other one thanks to C.2.

Proposition F.1. *Let $d>0$, for any set $X\subset\mathbb{R}^d$, h_X is convex.*

Proof. Let $(\lambda,\lambda')\in\mathcal{S}_X^2$ be two distributions in $\mathcal{S}_X$. If either $\Sigma_X(\lambda)$ or $\Sigma_X(\lambda')$ are not invertible, then for any $t\in[0,1]$ one has

$$h_X(t\lambda+(1-t)\lambda')\le th_X(\lambda)+(1-t)h_X(\lambda')=+\infty\ .$$

Otherwise, for $t\in[0,1]$, we define the positive definite matrix $\mathbf{Z}(t)\in\mathbb{R}^{d\times d}$ as follows:

$$\mathbf{Z}(t)=t\Sigma_X(\lambda)+(1-t)\Sigma_X(\lambda')\ .$$

Simple linear algebra (Petersen and Pedersen, 2012) yields

$$\frac{\partial\mathbf{Z}(t)^{-1}}{\partial t}=\mathbf{Z}(t)^{-1}\frac{\partial\mathbf{Z}(t)}{\partial t}\mathbf{Z}(t)^{-1}\ .$$

Using this result and the fact that $\partial^2\mathbf{Z}(t)/\partial t^2=0$, we obtain

$$\frac{\partial^2\mathbf{Z}(t)^{-1}}{\partial t^2}=2\mathbf{Z}(t)^{-1}\frac{\partial\mathbf{Z}(t)}{\partial t}\mathbf{Z}(t)^{-1}\frac{\partial\mathbf{Z}(t)}{\partial t}\mathbf{Z}(t)^{-1}\ .$$

Therefore, for any $x\in X$,

$$\begin{aligned}\frac{\partial^2x^\top\mathbf{Z}(t)^{-1}x}{\partial t^2}&=2x^\top\mathbf{Z}(t)^{-1}\frac{\partial\mathbf{Z}(t)}{\partial t}\mathbf{Z}(t)^{-1}\frac{\partial\mathbf{Z}(t)}{\partial t}\mathbf{Z}(t)^{-1}x\\&=2\left(\frac{\partial\mathbf{Z}(t)}{\partial t}\mathbf{Z}(t)^{-1}x\right)^\top\mathbf{Z}(t)^{-1}\left(\frac{\partial\mathbf{Z}(t)}{\partial t}\mathbf{Z}(t)^{-1}x\right)\\&\ge0\ ,\end{aligned}$$

which shows convexity for any fixed $x\in X$. The final results yields from the fact that h_X is a maximum over convex functions.

□

G. Additional experiment and information

We define the set of arms $\mathcal{X}\subset\mathbb{R}^5$ that is made of $|\mathcal{X}|=100$ node-arms randomly sampled from a multivariate 5-dimensional Gaussian distribution $\mathcal{N}(0,I)$ and then normalized so that $\|x\|=1$ for all $x\in\mathcal{X}$. In all the figures the results are averaged over 100 random repetitions of the experiments.

We propose to validate our insight and compute the evolution of $\|\mathrm{Var}\left(\mathbf{A}_1\right)\|$ for the three types of graphs (star, complete and circle) and different number of edges. The results are shown in Figure 2. One can notice that we retrieve the $O(m^2)$ dependence of the variance for the star graph, the $O(m\sqrt{m})$ for the complete graph and the linear dependence $O(m)$ for the circle graph.

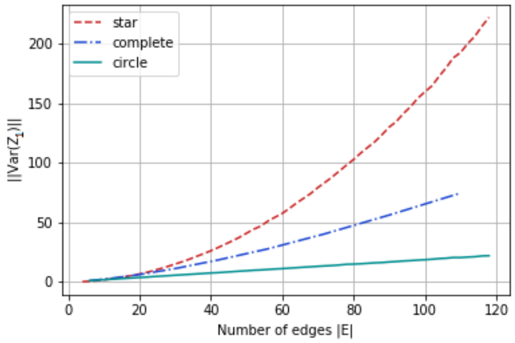

Figure 2. Evolution of the variance according to the number of edges and the type of graph (star, complete, circle), the variance being averaged over 100 repetitions.

Machine used for all the experiments. Intel(R) Xeon(R) CPU E5-2667 v4 @ 3.20GHz - 24 CPUs used.