# Selfish Robustness and Equilibria in Multi-Player Bandits

**Etienne Boursier** EBOURSIE@ENS-PARIS-SACLAY.FR
*Université Paris-Saclay, ENS Paris-Saclay, CNRS, Centre Borelli, Cachan, France.*

**Vianney Perchet** VIANNEY.PERCHET@NORMALESUP.ORG
*CREST, ENSAE Paris, Palaiseau, France; Criteo AI Lab, Paris, France*

**Editors:** Jacob Abernethy and Shivani Agarwal

## Abstract

Motivated by cognitive radios, stochastic multi-player multi-armed bandits gained a lot of interest recently. In this class of problems, several players simultaneously pull arms and encounter a collision – with 0 reward – if some of them pull the same arm at the same time. While the cooperative case where players maximize the collective reward (obediently following some fixed protocol) has been mostly considered, robustness to malicious players is a crucial and challenging concern. Existing approaches consider only the case of adversarial *jammers* whose objective is to blindly minimize the collective reward.

We shall consider instead the more natural class of selfish players whose incentives are to maximize their individual rewards, potentially at the expense of the social welfare. We provide the first algorithm robust to selfish players (a.k.a. Nash equilibrium) with a logarithmic regret, when the arm performance is observed. When collisions are also observed, *Grim Trigger* type of strategies enable some implicit communication-based algorithms and we construct robust algorithms in two different settings: the homogeneous (with a regret comparable to the centralized optimal one) and heterogeneous cases (for an adapted and relevant notion of regret). We also provide impossibility results when only the reward is observed or when arm means vary arbitrarily among players.

**Keywords:** Multi-Armed Bandits, Decentralized Algorithms, Cognitive Radio, Game Theory

## 1. Introduction

In the classical stochastic Multi Armed Bandit problem (MAB), a player repeatedly chooses among $K$ fixed actions (a.k.a. arms). After pulling arm $k \in [K] := \{1, \dots, K\}$, she receives a random reward in $[0, 1]$ of mean $\mu_k$. Her goal is to maximize her cumulative reward up to some horizon $T \in \mathbb{N}$. The performance of a pulling strategy (or algorithm) is assessed by the growth of *regret*, i.e., the difference between the highest possible expected cumulative reward and the actual cumulative reward. Since the means $\mu_k$ are unknown beforehand, the player trades off gathering information on under-sampled arms (exploration) vs. using her information (exploitation). Optimal solutions are known in the simplest model (Lai and Robbins, 1985; Agrawal, 1995; Auer et al., 2002). We refer to (Bubeck and Cesa-Bianchi, 2012; Lattimore and Szepesvári, 2018; Slivkins et al., 2019) for an extensive study of MAB. This simple model captures many sequential decisions problems including clinical trials (Thompson, 1933; Robbins, 1952) and online recommandation systems (Li et al., 2010) and has therefore known a large interest in the past decades.

Another classical application of MAB is cognitive radios (Jouini et al., 2009; Anandkumar et al., 2011). In this context, an arm corresponds to a channel on which a player decides to transmit and the reward is its transmission quality. A key feature of this model, is that it involves several players using channels simultaneously. If several players choose the same arm/channel at the same time, then they *collide* and receive a null reward. This setting remains somehow simple when a central agent controls simultaneously all players (Anantharam et al., 1987; Komiyama et al., 2015), which is far from being realistic. In reality, the problem is indeed completely decentralized: players are independent, anonymous and cannot communicate to each other. This requires the construction of new algorithms and the development of new techniques dedicated to this *multiplayer bandit* problem. Interestingly, there exist several variants of the base problem, depending on the assumption made on observations/feedback received (Avner and Mannor, 2014; Rosenski et al., 2016; Besson and Kaufmann, 2018; Lugosi and Mehrabian, 2018; Magesh and Veeravalli, 2019).

More precisely, when players systematically know whether or not they collide, this observation actually enables communication between players and a collective regret scaling as in the centralized case is possible, as observed recently (Boursier and Perchet, 2019; Proutiere and Wang, 2019). Using this idea, it is even possible to asymptotically reach the optimal assignment (Bistritz and Leshem, 2018; Tibrewal et al., 2019; Boursier et al., 2019) in the heterogeneous model where the performance of each arm differs among players (Kalathil et al., 2014; Avner and Mannor, 2015, 2018). Liu et al. (2019) considered the heterogeneous case, when arms also have preferences over players.

For the aforementioned result to hold, a crucial (yet sometimes only implicitly stated) assumption is that all players follow cautiously and meticulously some designed protocols and that none of them tries to free-ride the others by acting greedily, selfishly or maliciously. The concern of designing multiplayer bandit algorithms robust to such players has been raised (Attar et al., 2012), but only addressed under the quite restrictive assumption of adversarial players called *jammers*. Those try to perturb as much as possible the cooperative players (Wang et al., 2015; Sawant et al., 2018, 2019), even if this is extremely costly to them as well. Because of this specific objective, they end up using tailored strategies such as only attacking the top channels.

We focus instead on the construction of algorithms with "good" regret guarantees even if one (or actually more) selfish player does not follow the common protocol but acts strategically in order to manipulate the other players in the sole purpose of increasing her own payoff – maybe at the cost of other players. This concept appeared quite early in the cognitive radio literature (Attar et al., 2012), yet it is still not understood as robustness to selfish player is intrinsically different (and even non-compatible) with robustness to jammers, as shown in Section 2.2. In terms of game theory, we aim at constructing ($\varepsilon$-Nash) equilibria in this repeated game with partial observations.

The paper is organized as follows. Section 2 introduces notions and concepts of selfishness-robust multiplayer bandits and showcases reasons for the design of robust algorithms. Besides its state of the art regret guarantees when collisions are not directly observed, `Selfish-Robust MMAB`, presented in Section 3, is also robust to selfish players. In the more complex settings where only the reward is observed or the arm means vary among players, Section 4 shows that no algorithm can guarantee both a sublinear regret and selfish-robustness. The latter case is due to a more general result for random assignments. Instead of comparing the cumulated reward with the best collective assignment in the heterogeneous case, it is then necessary to compare it with a *good* and appropriate suboptimal assignment, leading to the new notion of *RSD-regret*.

When collisions are always observed, Section 5 proposes selfish-robust communication protocols. Thanks to this, an adaptation of the work of Boursier and Perchet (2019) is possible to provide a robust algorithm with a collective regret almost scaling as in the centralized case. In the heterogeneous case, this communication – along with other new deviation control and punishment protocols – is also used to provide a robust algorithm with a logarithmic RSD-regret.

Our contributions are thus diverse: on top of introducing notions of selfish-robustness, we provide robust algorithms with state of the art regret bounds (w.r.t. non-robust algorithms) in several settings. This is especially surprising when collisions are observed, since it leads to a near centralized regret. Moreover, we show that such algorithms can not be designed in harder settings. This leads to the new, adapted notion of RSD-regret in the heterogeneous case with selfish players and we also provide a *good* algorithm in this case. These results of robustness are even more intricate knowing they hold against any possible selfish strategy, in contrast to the known results for jammer robust algorithms.

## 2. Problem statement

In this section, we describe formally the model of multiplayer MAB and introduce concepts and notions of robustness to selfish players (or equilibria concepts).

### 2.1. Model

We denote the transmission qualities of the channels by $(X_k(t))_{1\leq k\leq K} \in [0,1]$, drawn i.i.d. according to $\nu_k$ of expectation $\mu_k$. In the following, arm means are assumed to be different and $\mu_{(i)}$ denotes the $i$-th largest mean, i.e., $\mu_{(1)} > \mu_{(2)} > \ldots > \mu_{(K)}$. At each round $t \in [T]$, all $M$ players simultaneously pull some arms, choice solely based only on their past own observations with $M \leq K$. We denote by $\pi^j(t)$ the arm played by player $j$, that generates the reward

$$r^j(t) \coloneqq X_{\pi^j(t)}(t) \cdot (1 - \eta_{\pi^j(t)}(t)),$$
$$\text{where } \eta_k(t) \coloneqq \mathbb{1}\left(\#\{j \in [M] \mid \pi^j(t) = k\} > 1\right) \text{ is the collision indicator.}$$

The performance of an algorithm is measured in terms of regret, i.e., the difference between the maximal expected reward and the algorithm cumulative reward after $T$ steps[1]:

$$R_T \coloneqq T \sum_{k=1}^{M} \mu_{(k)} - \sum_{t=1}^{T}\sum_{j=1}^{M} \mu_{\pi^j(t)}(t) \cdot (1 - \eta_{\pi^j(t)}(t)).$$

In multiplayer MAB, three different observation settings are considered.

**Full sensing:** each player observes both $\eta_{\pi^j(t)}(t)$ and $X_{\pi^j(t)}(t)$ at each round.

**Statistic sensing:** each player observes $X_{\pi^j(t)}(t)$ and $r^j(t)$ at each round, e.g., the players first sense the quality of a channel before trying to transmit on it.

**No sensing:** each player only observes $r^j(t)$ at each round.

Players are not able to directly communicate to each other, since it involves significant time and energy cost in practice. Some form of communication is still possible between players through observed collisions and has been widely used in recent literature (Boursier and Perchet, 2019; Boursier et al., 2019; Tibrewal et al., 2019; Proutiere and Wang, 2019).

1. As usual, the fact that the horizon $T$ is known is not crucial (Degenne and Perchet, 2016).

### 2.2. Considering selfish players

As mentioned in the introduction, the literature focused on adversarial malicious players, a.k.a. *jammers*, while considering selfish players instead of adversarial ones is as (if not more) crucial. These two concepts of malicious players are fundamentally different. Jamming-robust algorithms must stop pulling the best arm if it is being jammed. Against this algorithm, a selfish player could therefore pose as a jammer, always pull the best arm and be left alone on it most of the time. On the contrary, an algorithm robust to selfish players has to actually pull this best arm if jammed by some player in order to "punish" her so that she does not benefit from deviating from the collective strategy.

We first introduce some game theoretic concepts before defining notions of robustness. Each player $j$ follows an individual strategy (or algorithm) $s_j \in \mathcal{S}$ which determines her action at each round given her past observations. We denote by $(s_1, \dots, s_M) = s \in \mathcal{S}^M$ the strategy profile of all players and by $(s', s_{-j})$ the strategy profile given by $s$ except for the $j$-th player whose strategy is replaced by $s'$. Let $\mathrm{Rew}_T^j(s)$ be the cumulative reward of player $j$ when players play the profile $s$. As usual in game theory, we consider a single selfish player – even if the algorithms we propose are robust to several selfish players assuming $M$ is known beforehand (its initial estimation can easily be tricked by several players).

**Definition 1.** *A strategy profile $s \in \mathcal{S}^M$ is an $\varepsilon$-Nash equilibrium if for any $s' \in \mathcal{S}$ and $j \in [M]$:*

$$\mathbb{E}[\mathrm{Rew}_T^j(s', s_{-j})] \leq \mathbb{E}[\mathrm{Rew}_T^j(s)] + \varepsilon.$$

This simply states that a selfish player wins at most $\varepsilon$ by deviating from $s_j$. We now introduce a more restrictive property of stability that involves two points: if a selfish player still were to deviate, this would only incur a small loss to other players. Moreover, if the selfish player wants to incur some considerable loss to the collective players (e.g., she is adversarial), then she also has to incur a comparable loss to herself. Obviously, an $\varepsilon$-Nash equilibrium is $(0, \varepsilon)$-stable.

**Definition 2.** *A strategy profile $s \in \mathcal{S}^M$ is $(\alpha, \varepsilon)$-stable if for any $s' \in \mathcal{S}$, $l \in \mathbb{R}_+$ and $i, j \in [M]$:*

$$\mathbb{E}[\mathrm{Rew}_T^i(s', s_{-j})] \leq \mathbb{E}[\mathrm{Rew}_T^i(s)] - l \implies \mathbb{E}[\mathrm{Rew}_T^j(s', s_{-j})] \leq \mathbb{E}[\mathrm{Rew}_T^j(s)] + \varepsilon - \alpha l.$$

### 2.3. Limits of existing algorithms.

This section explains why existing algorithms are not robust to selfish players, i.e., are not even $o(T)$-Nash equilibria. Besides justifying the design of new appropriate algorithms, this provides some first insights on the way to achieve robustness.

**Communication between players.** Many recent algorithms rely on communication protocols between players to gather their statistics. Facing an algorithm of this kind, a selfish player would communicate fake statistics to the other players in order to keep the best arm for herself. In case of collision, the colliding player(s) remains unidentified, so a selfish player could modify *incognito* the statistics sent by other players, making them untrustworthy. A way to make such protocols robust to malicious players is proposed in Section 5. Algorithms relying on communication can then be adapted in the Full Sensing setting.

**Necessity of fairness** An algorithm is fair if all players asymptotically earn the same reward *a posteriori* and not only in expectation. As already noticed (Attar et al., 2012), fairness seems to be a significant criterion in the design of selfish-robust algorithms. Indeed, without fairness, a selfish player tries to always be the one with the largest reward *a posteriori*.

For example, against algorithms attributing an arm among the top-$M$ ones to each player (Rosenski et al., 2016; Besson and Kaufmann, 2018; Boursier and Perchet, 2019), a selfish player could easily rig the attribution to end with the best arm, largely increasing her individual reward. Other algorithms work on the basis of first come-first served (Boursier and Perchet, 2019). Players first explore and when they detect an arm as both optimal and available, they pull it forever. Such an algorithm is unfair and a selfish player could play more aggressively to end her exploration before the others and to commit on an arm, maybe at the risk of committing on a suboptimal one (but with high probability on the best arm). The risk taken by the early commit is small compared to the benefit of being the first committing player. As a consequence, these algorithms are not $o(T)$-Nash equilibria.

## 3. Statistic sensing setting

In the statistic sensing setting where $X_k$ and $r_k$ are observed at each round, the `Selfish-Robust MMAB` algorithm provides satisfying theoretical guarantees.

### 3.1. Description of `Selfish-Robust MMAB`

**Algorithm 1:** `Selfish-Robust MMAB`

**Input:** $T, \gamma_1 \coloneqq \frac{13}{14}, \gamma_2 \coloneqq \frac{16}{15}$
$\beta \leftarrow 39$; $\widehat{M}, t_m \leftarrow$ `EstimateM` $(\beta, T)$
Pull $k \sim \mathcal{U}(K)$ until round $\frac{\gamma_2}{\gamma_1} t_m$ `// first waiting room`
$j \leftarrow$ `GetRank` $(\widehat{M}, t_m, \beta, T)$ and pull $j$ until round $\left(\frac{\gamma_2}{\gamma_1^2 \beta^2 K^2} + \frac{\gamma_2^2}{\gamma_1^2}\right) t_m$
Run `Alternate Exploration` $(\widehat{M}, j)$ until $T$

A global description of `Selfish-Robust MMAB` is given by Algorithm 1. The pseudocodes of `EstimateM`, `GetRank` and `Alternate Exploration` are respectively given by Protocols 1, 2 and Algorithm 2 in Appendix A due to space constraints.

`EstimateM` and `GetRank` respectively estimate the number of players $M$ and attribute ranks in $[M]$ among the players. They form the initialization phase, while `Alternate Exploration` optimally balances between exploration and exploitation.

#### 3.1.1. INITIALIZATION PHASE

Let us first introduce the following quantities:

- $N_k^j(t) = \{t' \leq t \mid \pi^j(t') = k \text{ and } X_k(t') > 0\}$ are rounds when player $j$ observed $\eta_k$.
- $C_k^j(t) = \{t' \in N_k^j(t) \mid \eta_k(t') = 1\}$ are rounds when player $j$ observed a collision.
- $\hat{p}_k^j(t) = {\#C_k^j(t)}/{\#N_k^j(t)}$ is the empirical probability to collide on the arm $k$ for player $j$.

During the initialization, the players estimate $M$ with large probability as given by Lemma 1 in Appendix A.1. Players first pull uniformly at random in $[K]$. As soon as $\#N_k^j \geq n$ for any $k \in [K]$ and some fixed $n$, player $j$ ends the `EstimateM` protocol and estimates $\widehat{M}$ as the closest integer to $1 + \log(1 - \sum_k \hat{p}_k^j(t_M)/K) / \log(1 - \frac{1}{K})$. This estimation procedure is the same as the one of Rosenski et al. (2016), except for the following features:

i) Collisions indicators are not always observed, as we consider statistic sensing here. For this reason, the number of observations of $\eta_k$ is random. The stopping criterion $\min_k \#N_k^j(t) \geq n$ ensures that players don't need to know $\mu_{(K)}$ beforehand, but they also do not end `EstimateM` simultaneously. This is why a *waiting room* is needed, during which a player continues to pull uniformly at random to ensure that all players are still pulling uniformly at random if some player is still estimating $M$.

ii) The collision probability is not averaged over all arms, but estimated for each arm individually, then averaged. This is necessary for robustness as explained in Appendix A, despite making the estimation longer.

**Attribute ranks.** After this first procedure, players then proceed to a *Musical Chairs* (Rosenski et al., 2016) phase to attribute ranks among them as given by Lemma 2 in Appendix A.1. Players sample uniformly at random in $[M]$ and stop on an arm $j$ as soon as they observe a positive reward. The player's rank is then $j$ and only attributed to her. Here again, a *waiting room* is required to ensure that all players are either pulling uniformly at random or only pulling a specific arm (corresponding to their rank) during this procedure. During this second waiting room, a player thus pulls the arm corresponding to her rank.

#### 3.1.2. EXPLORATION/EXPLOITATION

After the initialization, players know $M$ and have different ranks. They enter the second phase, where they follow `Alternate Exploration`, inspired by Proutiere and Wang (2019). Player $j$ sequentially pulls arms in $\mathcal{M}^j(t)$, which is the ordered list of her $M$ best empirical arms, unless she has to pull her $M$-th best empirical arm. In that case, she instead chooses at random between actually pulling it or pulling an arm to explore (any arm not in $\mathcal{M}^j(t)$ with an upper confidence bound larger than the $M$-th best empirical mean, if there is any).

Since players proceed in a shifted fashion, they never collide when $\mathcal{M}^j(t)$ are the same for all $j$. Having different $\mathcal{M}^j(t)$ happens in expectation a constant (in $T$) amount of times, so that the contribution of collisions to the regret is negligible.

### 3.2. Theoretical results

This section provides theoretical guarantees of `Selfish-Robust MMAB`. Theorem 1 first presents guarantees in terms of regret. Its proof is given in Appendix A.2.1.

**Theorem 1.** *The collective regret of* `Selfish-Robust MMAB` *is bounded as*

$$\mathbb{E}[R_T] \leq M \sum_{k>M} \frac{\mu_{(M)} - \mu_{(k)}}{\mathrm{kl}(\mu_{(k)}, \mu_{(M)})} \log(T) + \mathcal{O}\left(\frac{MK^3}{\mu_{(K)}} \log(T)\right).$$

It can also be noted from Lemma 3 in Appendix A.2.1 that the regret due to `Alternate Exploration` is $M \sum_{k>M} \frac{\mu_{(M)}-\mu_{(k)}}{\mathrm{kl}(\mu_{(k)},\mu_{(M)})} \log(T) + o(\log(T))$, which is known to be optimal for algorithms using no collision information (Besson and Kaufmann, 2019). `Alternate Exploration` thus gives an optimal algorithm under this constraint, if $M$ is already known and ranks already attributed (as the $\mathcal{O}(\cdot)$ term in the regret is the consequence of their estimation).

On top of good regret guarantees, `Selfish-Robust MMAB` is robust to selfish behaviors as highlighted by Theorem 2 (whose proof is deterred to Appendix A.2.5).

**Theorem 2.** *Playing* `Selfish-Robust MMAB` *is an $\varepsilon$-Nash equilibrium and is $(\alpha, \varepsilon)$-stable*

$$\textit{with} \quad \varepsilon = \sum_{k>M} \frac{\mu_{(M)}-\mu_{(k)}}{\mathrm{kl}(\mu_{(k)},\mu_{(M)})} \log(T) + \mathcal{O}\left(\frac{\mu_{(1)}}{\mu_{(K)}} K^3 \log(T)\right) \quad \textit{and} \quad \alpha = \frac{\mu_{(M)}}{\mu_{(1)}}.$$

These points are proved for an *omniscient* selfish player (knowing all the parameters beforehand). This is a very strong assumption and a real player would not be able to win as much by deviating from the collective strategy. Intuitively, a selfish player would need to explore sub-optimal arms as given by the known individual lower bounds. However, a selfish player can actually decide to not explore but deduce the exploration of other players from collisions.

## 4. On harder problems

Following the positive results of the previous section (existence of robust algorithms) in the homogeneous case with statistical sensing, we now provide in this section impossibility results for both no sensing and heterogeneous cases. By showing its limitations, it also suggests a proper way to consider the heterogeneous problem in the presence of selfish players.

### 4.1. Hardness of no sensing setting

**Theorem 3.** *In the no sensing setting, there is no profile of strategy $s$ such that, for all problem parameters $(M, \boldsymbol{\mu})$, $\mathbb{E}[R_T] = o(T)$ and $s$ is an $\varepsilon(T)$-Nash equilibrium with $\varepsilon(T) = o(T)$.*

**Proof.** Consider a strategy $s$ verifying the first property and a problem instance $(M, \boldsymbol{\mu})$ where the selfish player only pulls the best arm. Let $\boldsymbol{\mu'}$ be the mean vector $\boldsymbol{\mu}$ where $\mu_{(1)}$ is replaced by 0. Then, because of the considered observation model, the cooperative players can not distinguish the two worlds $(M, \boldsymbol{\mu})$ and $(M-1, \boldsymbol{\mu'})$. Having a sublinear regret in the second world implies $o(T)$ pulls on the arm 1 for the cooperative players. So in the first world, the selfish player will have a reward in $\mu_{(1)}T - o(T)$, which is thus a linear improvement in comparison with following $s$ if $\mu_{(1)} > \mu_{(2)}$. ■

Theorem 3 is proved for a selfish players who knows the means $\boldsymbol{\mu}$ beforehand, as the notion of Nash equilibrium prevents against any possible strategy, which includes committing to an arm for the whole game. The knowledge of $\boldsymbol{\mu}$ is actually not needed, as a similar result holds for a selfish player committing to an arm chosen at random when the best arm is $K$ times better than the second one. The question of existence of robust algorithms remains yet open if we restrict selfish strategies to more *reasonable* algorithms.

### 4.2. Heterogeneous model

We consider the full sensing heterogeneous model, where player $j$ receives the reward $r^j(t) := X^j_{\pi^j(t)}(t)(1-\eta_{\pi^j(t)})$ at round $t$, with $X^j_k \overset{\text{i.i.d.}}{\sim} \nu^j_k$ of mean $\mu^j_k$. The arm means here vary among the players. This models that transmission quality depends on individual factors such as the localization.

#### 4.2.1. A FIRST IMPOSSIBILITY RESULT

**Theorem 4.** *If the regret is compared with the optimal assignment, there is no strategy $s$ such that, for all problem parameters $\boldsymbol{\mu}$, $\mathbb{E}[R_T] = o(T)$ and $s$ is an $\varepsilon(T)$-Nash equilibrium with $\varepsilon(T) = o(T)$.*

**Proof.** Assume $s$ satisfies these properties and consider a problem instance $\boldsymbol{\mu}$ such that the selfish player unique best arm $j_1$ has mean $\mu^j_{(1)} = 1/2$ and the difference between the optimal assignment utility and the utility of the best one assigning arm $j_1$ to $j$ is $1/3$.

Such an instance is of course possible. Consider a selfish player $j$ playing exactly the strategy $s_j$ but as if her reward vector $\boldsymbol{\mu^j}$ was actually $\boldsymbol{\mu'^j}$ where $\mu^j_{(1)}$ is replaced by 1 and all other $\mu^j_k$ by 0, i.e., she fakes a second world $\boldsymbol{\mu'}$ in which the optimal assignment gives her the arm $j_1$. In this case, the sublinear regret assumption of $s$ implies that player $j$ pulls $j_1$ a time $T - o(T)$, while in the true world, she would have pulled it $o(T)$ times. She thus earns an improvement at least $(\mu^j_{(1)} - \mu^j_{(2)})T - o(T)$ w.r.t. playing $s_j$, contradicting the Nash equilibrium assumption. ■

#### 4.2.2. RANDOM ASSIGNMENTS

We now take a step back and describe "relevant" allocation procedures for the heterogeneous case, when the vector of means $\boldsymbol{\mu^j}$ is already known by player $j$.

An assignment is *symmetric* if, when $\boldsymbol{\mu^j} = \boldsymbol{\mu^i}$, players $i$ and $j$ get the same **expected** utility, i.e., no player is *a priori* favored[2]. It is *strategyproof* if being truthful is a dominant strategy for any player and *Pareto optimal* if the social welfare (sum of utilities) can not be improved without hurting any player. Theorem 4 is a consequence of Theorem 5 below.

**Theorem 5 (Zhou 1990).** *For $M \geq 3$, there is no symmetric, Pareto optimal and strategyproof random assignment algorithm.*

Liu et al. (2019) circumvent this assignment problem with player-preferences for arms. Instead of assigning a player to a contested arm, the latter decides who gets to pull it, following its preferences.

In the case of random assignment, Abdulkadiroglu and Sonmez (1998) proposed the Random Serial Dictatorship (RSD) algorithm, which is symmetric and strategyproof. The algorithm is rather simple: pick uniformly at random an ordering of the $M$ players. Following this order, the first player picks her preferred arm, the second one her preferred remaining arm and so on. Svensson (1999) justified the choice of RSD for symmetric strategyproof assignment algorithms. Adamczyk et al. (2014) recently studied efficiency ratios of such assignments: if $U_{\max}$ denotes the expected social welfare of the optimal assignment, the expected social welfare of RSD is greater than $U_{\max}^2/eM$ while no strategyproof algorithm can guarantee more than $U_{\max}^2/M$. As a consequence, RSD is optimal up to a (multiplicative) constant and will serve as a benchmark in the remaining.

Indeed, instead of defining the regret in comparison with the optimal assignment as done in the classical heterogeneous multiplayer bandits, we are going to define it with respect to RSD to incorporate strategy-proofness constraints. Formally, the RSD-regret is defined as:

$$R_T^{\text{RSD}} := T\mathbb{E}_{\sigma\sim\mathcal{U}(\mathfrak{S}_M)}\left[\sum_{k=1}^{M} \mu^{\sigma(k)}_{\pi_\sigma(k)}\right] - \sum_{t=1}^{T}\sum_{j=1}^{M} \mu^j_{\pi^j(t)}(t)\cdot(1-\eta_{\pi^j(t)}(t)),$$

with $\mathfrak{S}_M$ the set of permutations over $[M]$ and $\pi_\sigma(k)$ the arm attributed by RSD to player $\sigma(k)$ when the order of dictators is $(\sigma(1), \dots, \sigma(M))$. Mathematically, $\pi_\sigma$ is defined by:

$$\pi_\sigma(1) = \operatorname*{arg\,max}_{l\in[M]} \mu_l^{\sigma(1)} \quad \text{and} \quad \pi_\sigma(k+1) = \operatorname*{arg\,max}_{\substack{l\in[M]\\ l\notin\{\pi_\sigma(l')\ |\ l'\leq k\}}} \mu_l^{\sigma(k+1)}.$$

## 5. Full sensing setting

This section focuses on the full sensing setting, where both $\eta_k(t)$ and $X_k(t)$ are always observed as we proved impossibility results for more complex settings. As mentioned before, recent algorithms leverage the observation of collisions to enable some communication between players by forcing them. Some of these communication protocols can be modified to allow robust communication. This section is structured as follows. First, insights on two new protocols are given for robust communications. Second, a robust adaptation of SIC-MMAB is given, based on these two protocols. Third, they can also be used to reach a logarithmic RSD-regret in the heterogeneous case.

2. The concept of fairness introduced above is stronger, as no player should be *a posteriori* favored.

### 5.1. Making communication robust

To have robust communication, two new complementary protocols are needed. The first one allows to send messages between players and to detect when they have been corrupted by a malicious player. If this has been the case, the players then use the second protocol to proceed to a collective punishment, which forces every player to suffer a considerable loss for the remaining of the game. Such punitive strategies are called "Grim Trigger" in game theory and are used to deter defection in repeated games (Friedman, 1971; Axelrod and Hamilton, 1981; Fudenberg and Maskin, 2009).

#### 5.1.1. BACK AND FORTH MESSAGING

Communication protocols in the collision sensing setting usually rely on the fact that collision indicators can be seen as bits sent from a player to another one as follows. If player $i$ sends a binary message $m_{i\to j} = (1, 0, \dots, 0, 1)$ to player $j$ during a predefined time window, she proceeds to the sequence of pulls $(j, i, \dots, i, j)$, meaning she purposely collides with $j$ to send a 1 bit (reciprocally, not colliding corresponds to a 0 bit). A malicious player trying to corrupt a message can only create new collisions, i.e., replace zeros by ones. The key point is that the inverse operation is not possible.

If player $j$ receives the (potentially corrupted) message $\widehat{m}_{i\to j}$, she repeats it to player $i$. This second message can also be corrupted by the malicious player and player $i$ receives $\widetilde{m}_{i\to j}$. However, since the only possible operation is to replace zeros by ones, there is no way to transform back $\widehat{m}_{i\to j}$ to $m_{i\to j}$ if the first message had been corrupted. The player $i$ then just has to compare $\widetilde{m}_{i\to j}$ with $m_{i\to j}$ to know whether or not at least one of the two messages has been corrupted. We call this protocol *back and forth* communication.

In the following, other malicious communications are possible. Besides sending false information (which is managed differently), a malicious player can send different statistics to the others, while they need to have the exact same statistics. To overcome this issue, players will send to each other statistics sent to them by any player. If two players have received different statistics by the same player, at least one of them automatically realizes it.

#### 5.1.2. COLLECTIVE PUNISHMENT

The back and forth protocol detects if a malicious player interfered in a communication and, in that case, a collective punishment is triggered (to deter defection). The malicious player is yet unidentified and can not be specifically targeted. The punishment thus guarantees that the average reward earned by any player is smaller than the average reward of the algorithm, $\bar{\mu}_M \coloneqq \frac{1}{M}\sum_{k=1}^{M} \mu_{(k)}$.

A naive way to *punish* is to pull all arms uniformly at random. The selfish player then gets the reward $(1-1/K)^{M-1}\mu_{(1)}$ by pulling the best arm, which can be larger than $\bar{\mu}_M$. A good punishment should therefore pull arms more often the better they are.

During the punishment, players pull each arm $k$ with probability $1 - \big(\gamma \frac{\sum_{l=1}^{M} \widehat{\mu}^j_{(l)}(t)}{M\widehat{\mu}^j_k(t)}\big)^{\frac{1}{M-1}}$ at least, where $\gamma = (1 - 1/K)^{M-1}$. Such a strategy is possible as shown by Lemma 13 in Appendix B. Assuming the arms are correctly estimated, i.e., the expected reward a selfish player gets by pulling $k$ is approximately $\mu_k(1 - p_k)^{M-1}$, with $p_k = \max\Big(1 - \big(\gamma\frac{\bar{\mu}_M}{\mu_k}\big)^{\frac{1}{M-1}}, 0\Big)$.

If $p_k = 0$, then $\mu_k$ is smaller than $\gamma\bar{\mu}_M$ by definition; otherwise, it necessarily holds that $\mu_k(1 - p_k)^{M-1} = \gamma\bar{\mu}_M$. As a consequence, in both cases, the selfish player earns at most $\gamma\bar{\mu}_M$, which involves a relative positive decrease of $1-\gamma$ in reward w.r.t. following the cooperative strategy. More details on this protocol are given by Lemma 21 in Appendix C.3.

### 5.2. Homogeneous case: `SIC-GT`

In the homogeneous case, these two protocols can be incorporated in the SIC-MMAB algorithm of Boursier and Perchet (2019) to provide `SIC-GT`, which is robust to selfish behaviors and still ensures a regret comparable to the centralized lower bound.

Boursier et al. (2019) recently improved the communication protocol by choosing a leader and communicating all the information only to this leader. A malicious player would do anything to be the leader. `SIC-GT` avoids such a behavior by choosing two leaders who either agree or trigger the punishment. More generally with $n+1$ leaders, this protocol is robust to $n$ selfish players. The detailed algorithm is given by Algorithm 3 in Appendix C.1.

**Initialization.** The original initialization phase of SIC-MMAB has a small regret term, but it is not robust. During the initialization, the players here pull uniformly at random to estimate $M$ as in `Selfish-Robust MMAB` and then attribute ranks the same way. The players with ranks 1 and 2 are then leaders. Since the collision indicator is always observed here, this estimation can be done in an easier and better way. The observation of $\eta_k$ also enables players to remain synchronized after this phase as its length does not depend on unknown parameters.

**Exploration and Communication.** Players alternate between exploration and communication once the initialization is over. During the $p$-th exploration phase, each arm still requiring exploration is pulled $2^p$ times by every player in a collisionless fashion. Players then communicate to each leader their empirical means in binary after every exploration phase, using the back and forth trick explained in Section 5.1.1. Leaders then check that their information match. If some undesired behavior is detected, a collective punishment is triggered.

Otherwise, the leaders determine the sets of optimal/suboptimal arms and send them to everyone. To prevent the selfish player from sending fake statistics, the leaders gather the empirical means of all players, except the extreme ones (largest and smallest) for every arm. If the selfish player sent outliers, they are thus cut out from the collective estimator, which is thus the average of $M-2$ individual estimates. This estimator can be biased by the selfish player, but a concentration bound given by Lemma 17 in Appendix C.2.1 still holds.

**Exploitation.** As soon as an arm is detected as optimal, it is pulled until the end. To ensure fairness of `SIC-GT`, players will actually rotate over all the optimal arms so that none of them is favored. This point is thoroughly described in Appendix C.1. Theorem 6, proved in Appendix C, gives theoretical results for `SIC-GT`.

**Theorem 6.** *Define* $\alpha = \frac{1-(1-1/K)^{M-1}}{2}$ *and assume* $M \geq 3$.

1. *The collective regret of* `SIC-GT` *is bounded as*

$$\mathbb{E}[R_T] \leq \mathcal{O}\Big(\sum_{k>M} \frac{\log(T)}{\mu_{(M)} - \mu_{(k)}} + MK^2\log(T) + M^2K\log^2\Big(\frac{\log(T)}{(\mu_{(M)} - \mu_{(M+1)})^2}\Big)\Big).$$

2. *Playing* `SIC-GT` *is an* $\varepsilon$*-Nash equilibrium and is* $(\alpha, \varepsilon)$*-stable with*

$$\varepsilon = \mathcal{O}\Big(\sum_{k>M} \frac{\log(T)}{\mu_{(M)} - \mu_k} + K^2\log(T) + MK\log^2\Big(\frac{\log(T)}{(\mu_{(M)} - \mu_{(M+1)})^2}\Big) + \frac{K\log(T)}{\alpha^2\mu_{(K)}}\Big).$$

### 5.3. Semi-heterogeneous case: RSD-GT

The punishment strategies described above can not be extended to the heterogeneous case, as the relevant probability of choosing each arm would depend on the preferences of the malicious player which are unknown (even her identity might not be discovered). Moreover, as already explained in the homogeneous case, pulling each arm uniformly at random is not an appropriate punishment strategy[3]. We therefore consider the $\delta$-heterogeneous setting, which allows punishments for small values of $\delta$ as given by Lemma 24 in Appendix D.3. The heterogeneous model was justified by the fact that transmission quality depends on individual factors such as localization. The $\delta$-heterogeneous assumption relies on the idea that such individual factors are of a different order of magnitude than global factors (as the availability of a channel). As a consequence, even if arm means differ from player to player, these variations remain relatively small.

**Definition 3.** *The setting is $\delta$-heterogeneous if there exists $\{\mu_k; k \in [K]\}$ such that for all $j$ and $k$, $\mu_k^j \in [(1-\delta)\mu_k, (1+\delta)\mu_k]$.*

In the semi-heterogeneous full sensing setting, RSD-GT provides a robust, logarithmic RSD-regret algorithm. Its complete description is given by Algorithm 4 in Appendix D.1.

#### 5.3.1. ALGORITHM DESCRIPTION

RSD-GT starts with the exact same initialization as SIC-GT to estimate $M$ and attribute ranks among the players. The time is then divided into superblocks which are divided into $M$ blocks. During the $j$-th block of a superblock, the dictators ordering[4] is $(j, \ldots, M, 1, \ldots, j-1)$. Moreover, only the $j$-th player can send messages during this block.

**Exploration.** The exploring players pull sequentially all the arms. Once player $j$ knows her $M$ best arms and their ordering, she waits for a block $j$ to initiate communication.

**Communication.** Once a player starts a communication block, she proceeds in three successive steps as follows:

1. she first collides with all players to signal the beginning of a communication block. The other players then enter a listening state, ready to receive messages.
2. She then sends to every player her ordered list of $M$ best arms. Each player then repeats this list to detect the potential intervention of a malicious player.
3. Finally, any player who detected the intervention of a malicious player signals to everyone the beginning of a collective punishment.

After a communication block $j$, every one knows the preferences order of player $j$, who is now in her exploitation phase, unless a punishment protocol has been started.

**Exploitation.** While exploiting, player $j$ knows the preferences of all other exploiting players. Thanks to this, she can easily compute the arms attributed by the RSD algorithm between the exploiting players, given the dictators ordering of the block.

Moreover, as soon as she collides in the beginning of a block while not intended (by her), this means an exploring player is starting a communication block. The exploiting player then starts listening to the arm preferences of the communicating player.

3. Unless in the specific case where $\mu_{(1)}^j(1-1/K)^{M-1} < \frac{1}{M}\sum_{k=1}^M \mu_{(k)}^j$.
4. The ordering is actually $(\sigma(j), \ldots, \sigma(j-1))$ where $\sigma(j)$ is the player with rank $j$ after the initialization. For sake of clarity, this consideration is omitted here.

#### 5.3.2. THEORETICAL GUARANTEES

Here are some insights to understand how RSD-GT reaches the utility of the RSD algorithm, which are rigorously detailed by Lemma 25 in Appendix D.3. With no malicious player, the players ranks given by the initialization provide a random permutation $\sigma \in \mathfrak{S}_M$ of the players and always considering the dictators ordering $(1, \ldots, M)$ would lead to the expected reward of the RSD algorithm. However, a malicious player can easily rig the initialization to end with rank 1. In that case, she largely improves her individual reward w.r.t. following the cooperative strategy.

To avoid such a behavior, the dictators ordering should rotate over all permutations of $\mathfrak{S}_M$, so that the rank of the player has no influence. However, this leads to an undesirable combinatorial $M!$ dependency of the regret. RSD-GT instead rotates over the dictators ordering $(j, \ldots, M, 1, \ldots, j-1)$ for all $j \in [M]$. If we note $\sigma_0$ the $M$-cycle $(1 \ldots M)$, the considered permutations during a superblock are of the form $\sigma \circ \sigma_0^{-m}$ for $m \in [M]$. The malicious player $j$ can only influence the distribution of $\sigma^{-1}(j)$: assume w.l.o.g. that $\sigma(1) = j$. The permutation $\sigma$ given by the initialization then follows the uniform distribution over $\mathfrak{S}_M^{j \to 1} = \{\sigma \in \mathfrak{S}_M \mid \sigma(1) = j\}$. But then, for any $m \in [M]$, $\sigma \circ \sigma_0^{-m}$ has a uniform distribution over $\mathfrak{S}_M^{j \to 1+m}$. In average over a superblock, the induced permutation still has a uniform distribution over $\mathfrak{S}_M$. So the malicious player has no interest in choosing a particular rank during the initialization, making the algorithm robust.

Thanks to this remark and robust communication protocols, RSD-GT possesses theoretical guarantees given by Theorem 7 (whose proof is deterred to Appendix D).

**Theorem 7.** *Consider the $\delta$-heterogeneous setting and define* $r = \frac{1-\left(\frac{1+\delta}{1-\delta}\right)^2(1-1/K)^{M-1}}{2}$ *and* $\Delta = \min_{(j,k)\in[M]^2} \mu^j_{(k)} - \mu^j_{(k+1)}$.

1. *The RSD-regret of* RSD-GT *is bounded as:* $\mathbb{E}[R_T] \leq \mathcal{O}\big(MK\Delta^{-2}\log(T) + MK^2\log(T)\big)$.
2. *If $r > 0$, playing* RSD-GT *is an $\varepsilon$-Nash equilibrium and is $(\alpha, \varepsilon)$-stable with*
    - $\varepsilon = \mathcal{O}\Big(\frac{K\log(T)}{\Delta^2} + K^2\log(T) + \frac{K\log(T)}{(1-\delta)r^2\mu_{(K)}}\Big)$,
    - $\alpha = \min\left(r\left(\frac{1+\delta}{1-\delta}\right)^3 \frac{\sqrt{\log(T)}-4M}{\sqrt{\log(T)}+4M}, \quad \frac{\Delta}{(1+\delta)\mu_{(1)}}, \quad \frac{(1-\delta)\mu_{(M)}}{(1+\delta)\mu_{(1)}}\right)$.

## 6. Conclusion

We introduced notions of robustness to selfish players and provided impossibility results in hard settings. With statistic sensing, Selfish-Robust MMAB gives a rather simple robust and efficient algorithm, besides being optimal among the class of algorithms using no collision information. On the other hand when collisions are observed, robust algorithms relying on communication through collisions are possible. Thanks to this, even selfish-robust algorithms can achieve near centralized regret in the homogeneous case, which is not intuitive at first sight. In the heterogeneous case, a new adapted notion of regret is introduced and RSD-GT achieves a good performance with respect to it.

RSD-GT heavily relies on collision observations and future work should focus on designing a comparable algorithm in both performance and robustness without this feature. The topic of robustness to selfish players in multiplayer bandits still remains largely unexplored and leaves open many directions for future work. In particular, punishment protocols do not seem possible for general heterogeneous settings and the existence of robust algorithms for any heterogeneous setting remains open. Also, stronger notions of equilibrium can be considered such as perfect subgame equilibrium.

## References

A. Abdulkadiroglu and T. Sonmez. Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica*, 66(3):689, 1998.

M. Adamczyk, P. Sankowski, and Q. Zhang. Efficiency of truthful and symmetric mechanisms in one-sided matching. In *International Symposium on Algorithmic Game Theory*, pages 13–24. Springer, 2014.

R. Agrawal. Sample mean based index policies with o(log n) regret for the multi-armed bandit problem. *Advances in Applied Probability*, 27(4):1054–1078, 1995.

A. Anandkumar, N. Michael, A. K. Tang, and A. Swami. Distributed algorithms for learning and cognitive medium access with logarithmic regret. *IEEE Journal on Selected Areas in Communications*, 29(4):731–745, 2011.

V. Anantharam, P. Varaiya, and J. Walrand. Asymptotically efficient allocation rules for the multi-armed bandit problem with multiple plays-part i: I.i.d. rewards. *IEEE Transactions on Automatic Control*, 32(11):968–976, 1987.

A. Attar, H. Tang, A. V. Vasilakos, F. R. Yu, and V. C. M. Leung. A survey of security challenges in cognitive radio networks: Solutions and future research directions. *Proceedings of the IEEE*, 100 (12):3172–3186, 2012.

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

O. Avner and S. Mannor. Concurrent bandits and cognitive radio networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 66–81. Springer, 2014.

O. Avner and S. Mannor. Learning to coordinate without communication in multi-user multi-armed bandit problems. *arXiv preprint arXiv:1504.08167*, 2015.

O. Avner and S. Mannor. Multi-user communication networks: A coordinated multi-armed bandit approach. *arXiv preprint arXiv:1808.04875*, 2018.

R. Axelrod and W. D. Hamilton. The evolution of cooperation. *science*, 211(4489):1390–1396, 1981.

Q. Berthet and V. Perchet. Fast rates for bandit optimization with upper-confidence Frank-Wolfe. In *Advances in Neural Information Processing Systems*, pages 2225–2234, 2017.

L. Besson and E. Kaufmann. Multi-Player Bandits Revisited. In *Algorithmic Learning Theory*, Lanzarote, Spain, 2018.

L. Besson and E. Kaufmann. Lower bound for multi-player bandits: Erratum for the paper multi-player bandits revisited, 2019.

I. Bistritz and A. Leshem. Distributed multi-player bandits-a game of thrones approach. In *Advances in Neural Information Processing Systems*, pages 7222–7232. 2018.

E. Boursier and V. Perchet. SIC-MMAB: synchronisation involves communication in multiplayer multi-armed bandits. *NeurIPS*, 2019.

E. Boursier, E. Kaufmann, A. Mehrabian, and V. Perchet. A practical algorithm for multiplayer bandits when arm means vary among players. *AISTATS*, 2019.

S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

N. Cesa-Bianchi, T. Cesari, and V. Perchet. Dynamic pricing with finitely many unknown valuations. In *Algorithmic Learning Theory*, pages 247–273, 2019.

R. Combes, S. Magureanu, A. Proutiere, and C. Laroche. Learning to rank: Regret lower bounds and efficient algorithms. *ACM SIGMETRICS Performance Evaluation Review*, 43(1):231–244, 2015.

R. Degenne and V. Perchet. Anytime optimal algorithms in stochastic multi-armed bandits. In *International Conference on Machine Learning*, pages 1587–1595, 2016.

J. W. Friedman. A non-cooperative equilibrium for supergames. *The Review of Economic Studies*, 38(1):1–12, 1971.

D. Fudenberg and E. Maskin. The folk theorem in repeated games with discounting or with incomplete information. In *A Long-Run Collaboration On Long-Run Games*, pages 209–230. World Scientific, 2009.

A. Garivier and O. Cappé. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Conference On Learning Theory*, pages 359–376, 2011.

W. Jouini, D. Ernst, C. Moy, and J. Palicot. Multi-armed bandit based policies for cognitive radio's decision making issues. In *2009 3rd International Conference on Signals, Circuits and Systems (SCS)*, 2009.

D. Kalathil, N. Nayyar, and R. Jain. Decentralized learning for multiplayer multiarmed bandits. *IEEE Transactions on Information Theory*, 60(4):2331–2345, 2014.

J. Komiyama, J. Honda, and H. Nakagawa. Optimal regret analysis of thompson sampling in stochastic multi-armed bandit problem with multiple plays. In *International Conference on Machine Learning*, pages 1152–1161, 2015.

T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

T. Lattimore and C. Szepesvári. Bandit algorithms. *preprint*, page 28, 2018.

L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

L.T. Liu, H. Mania, and M.I. Jordan. Competing bandits in matching markets. *arXiv preprint arXiv:1906.05363*, 2019.

G. Lugosi and A. Mehrabian. Multiplayer bandits without observing collision information. *arXiv preprint arXiv:1808.08416*, 2018.

A. Magesh and V. Veeravalli. Multi-player multi-armed bandits with non-zero rewards on collisions for uncoordinated spectrum access. *arXiv preprint arXiv:1910.09089*, 2019.

A. Proutiere and P. Wang. An optimal algorithm in multiplayer multi-armed bandits, 2019.

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

J. Rosenski, O. Shamir, and L. Szlak. Multi-player bandits–a musical chairs approach. In *International Conference on Machine Learning*, pages 155–163, 2016.

S. Sawant, M. K. Hanawal, S. Darak, and R. Kumar. Distributed learning algorithms for coordination in a cognitive network in presence of jammers. In *2018 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pages 1–8. IEEE, 2018.

S. Sawant, R. Kumar, M. K. Hanawal, and S. J. Darak. Learning to coordinate in a decentralized cognitive radio network in presence of jammers. *IEEE Transactions on Mobile Computing*, 2019.

A. Slivkins et al. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019.

L.-G. Svensson. Strategy-proof allocation of indivisible goods. *Social Choice and Welfare*, 16(4): 557–567, 1999.

W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.

H. Tibrewal, S. Patchala, M. K. Hanawal, and S. J. Darak. Distributed learning and optimal assignment in multiplayer heterogeneous networks. In *IEEE INFOCOM 2019*, pages 1693–1701, 2019.

Q. Wang, K. Ren, P. Ning, and S. Hu. Jamming-resistant multiradio multichannel opportunistic spectrum access in cognitive radio networks. *IEEE Transactions on Vehicular Technology*, 65(10): 8331–8344, 2015.

L. Zhou. On a conjecture by Gale about one-sided matching problems. *Journal of Economic Theory*, 52(1):123–135, 1990.

## Appendix A. Supplementary material for Section 3

This section provides a complete description of `Selfish-Robust MMAB` and the proofs of Theorems 1 and 2.

### A.1. Thorough description of `Selfish-Robust MMAB`

In addition to Section 3, the pseudocodes of `EstimateM`, `GetRank` and `Alternate Exploration` are given here. The following Protocol 1 describes the estimation of $M$ using the notations introduced in Section 3.1.1.

**Protocol 1:** `EstimateM`

**Input:** $\beta, T$
$t_m \leftarrow 0$
**while** $\min_k \#N_k^j(t) < \beta^2 K^2 \log(T)$ **do**
  Pull $k \sim \mathcal{U}(K)$; Update $\#N_k^j(t)$ and $\#C_k^j(t)$ ; $t_m \leftarrow t_m + 1$
**end**
$\widehat{M} \leftarrow 1 + \text{round}\left(\frac{\log\left(1 - \frac{1}{K}\sum_k \hat{p}_k^j(t_M)\right)}{\log\left(1-\frac{1}{K}\right)}\right)$ // round$(x)$ = closest integer to $x$
**Return** $\widehat{M}, t_m$

Since the duration $t_m^j$ of `EstimateM` for player $j$ is random and differs between players, each player continues sampling uniformly at random until $\frac{\gamma_2}{\gamma_1} t_m^j$, with $\gamma_1 = \frac{13}{14}$ and $\gamma_2 = \frac{16}{15}$. Thanks to this additional *waiting room*, Lemma 1 below guarantees that all players are sampling uniformly at random until at least $t_m^j$ for any $j$.

The estimation of $M$ here tightly estimates the probability to collide individually for each arm. This restriction provides an additional $M$ factor in the length of this phase in comparison with (Rosenski et al., 2016), where the probability to collide is globally estimated. This is however required because of the Statistic Sensing, but if $\eta_k$ was always observed, then the protocol from Rosenski et al. (2016) would be robust.

Indeed, if we directly estimated the global probability to collide, the selfish player could pull only the best arm. The number of observations of $\eta_k$ is larger on this arm, and the estimated probability to collide would thus be positively biased because of the selfish player.

Afterwards, ranks in $[M]$ are attributed to players by sampling uniformly at random in $[M]$ until observing no collision, as described in Protocol 2. For the same reason, a waiting room is added to guarantee that all players end this protocol with different ranks.

The following quantities are used to describe `Alternate Exploration` in Algorithm 2:

- $\mathcal{M}^j(t) = \left(l_1^j(t), \dots, l_M^j(t)\right)$ is the list of the empirical $M$ best arms for player $j$ at round $t$. It is updated only each $M$ rounds and ordered according to the index of the arms, i.e., $l_1^j(t) < \ldots < l_M^j(t)$.
- $\widehat{m}^j(t)$ is the empirical $M$-th best arm for player $j$ at round $t$.
- $b_k^j(t) = \sup\{q \geq 0 \mid T_k^j(t)\text{kl}(\widehat{\mu}_k^j(t), q) \leq f(t)\}$ is the kl-UCB index of the arm $k$ for player $j$ at round $t$, where $f(t) = \log(t) + 4\log(\log(t))$, $T_k^j(t)$ is the number of times player $j$ pulled $k$ and $\widehat{\mu}_k^j$ is the empirical mean.

**Protocol 2:** `GetRank`

**Input:** $\widehat{M}, t_m^j, \beta, T$
$n \leftarrow \beta^2 K^2 \log(T)$ and $j \leftarrow -1$
**for** $t_m^j \log(T)/(\gamma_1 n)$ *rounds* **do**
  **if** $j = -1$ **then**
    Pull $k \sim \mathcal{U}(\widehat{M})$;  **if** $r_k(t) > 0$ **then** $j \leftarrow k$   `// no collision`
  **else** Pull $j$
**end**
**Return** $j$

**Algorithm 2:** `Alternate Exploration`

**Input:** $M, j$
**if** $t = 0 \pmod M$ **then** Update $\hat{\mu}^j(t), b^j(t), \widehat{m}^j(t)$ and $\mathcal{M}^j(t) = (l_1, \ldots, l_M)$
$\pi \leftarrow t + j \pmod M + 1$
**if** $l_\pi \neq \widehat{m}^j(t)$ **then** Pull $l_\pi$   `// exploit the` $M-1$ `best empirical arms`
**else**
  $\mathcal{B}^j(t) = \{k \notin \mathcal{M}^j(t) \mid b_k^j(t) \geq \widehat{\mu}^j_{\widehat{m}^j(t)}(t)\}$   `// arms to explore`
  **if** $\mathcal{B}^j(t) = \emptyset$ **then** Pull $l_\pi$
  **else** Pull $\begin{cases} l_\pi \text{ with proba } 1/2 \\ k \text{ chosen uniformly at random in } \mathcal{B}^j(t) \text{ otherwise} \end{cases}$   `// explore`
**end**

### A.2. Proofs of Section 3

Let us define $\alpha_k := \mathbb{P}(X_k(t) > 0) \geq \mu_k$, $\gamma_1 = \frac{13}{14}$ and $\gamma_2 = \frac{16}{15}$.

#### A.2.1. REGRET ANALYSIS

This section aims at proving Theorem 1. This proof is divided in several auxiliary lemmas given below. First, the regret can be decomposed as follows:

$$R_T = R^{\text{init}} + R^{\text{explo}}, \tag{1}$$

$$\text{where } \begin{cases} R^{\text{init}} = T_0 \sum_{k=1}^{M} \mu_{(k)} - \mathbb{E}_\mu \Big[ \sum_{t=1}^{T_0} \sum_{j=1}^{M} r^j(t) \Big] \text{ with } T_0 = \left( \frac{\gamma_2}{\gamma_1^2 \beta^2 K^2} + \frac{\gamma_2^2}{\gamma_1^2} \right) \max_j t_m^j, \\ R^{\text{explo}} = (T - T_0) \sum_{k=1}^{M} \mu_{(k)} - \mathbb{E}_\mu \Big[ \sum_{t=T_0+1}^{T} \sum_{j=1}^{M} r^j(t) \Big]. \end{cases}$$

Lemma 1 first gives guarantees on the `EstimateM` protocol. Its proof is given in Appendix A.2.2.

**Lemma 1.** *If $M-1$ players run* `EstimateM` *with $\beta \geq 39$, followed by a waiting room until $\frac{\gamma_2}{\gamma_1} t_m^j$, then regardless of the strategy of the remaining player, with probability larger than $1 - \frac{6KM}{T}$, for any player:*

$$\widehat{M}^j = M \text{ and } \frac{t_m^j \alpha_{(K)}}{K} \in [\gamma_1 n, \gamma_2 n],$$

*where $n = \beta^2 K^2 \log(T)$.*

When $\widehat{M}^j = M$ and $\frac{t_m^j \alpha_{(K)}}{K} \in [\gamma_1 n, \gamma_2 n]$ for any cooperative player $j$, we say that the estimation phase is **successful**.

**Lemma 2.** *Conditioned on the success of the estimation phase, with probability $1 - \frac{M}{T}$, all the cooperative players end* `GetRank` *with different ranks $j \in [M]$, regardless of the behavior of other players.*

The proof of Lemma 2 is given in Appendix A.2.3. If the estimation is successful and all players end `GetRank` with different ranks $j \in [M]$, the initialization is said successful.

Using the same arguments as Proutiere and Wang (2019), the collective regret of the `Alternate Exploration` phase can be shown to be $M \sum_{k>M} \frac{\mu_{(M)} - \mu_{(k)}}{\mathrm{kl}(\mu_{(M)}, \mu_{(k)})} \log(T) + o(\log(T))$. This result is given by Lemma 3, whose proof is given in Appendix A.2.4.

**Lemma 3.** *If all players follow* `Selfish-Robust MMAB`*:*

$$\mathbb{E}[R^{\text{explo}}] \leq M \sum_{k>M} \frac{\mu_{(M)} - \mu_{(k)}}{\mathrm{kl}(\mu_{(M)}, \mu_{(k)})} \log(T) + o(\log(T)).$$

**Proof of Theorem 1.** Thanks to Lemma 3, the total regret is bounded by

$$M \sum_{k>M} \frac{\mu_{(M)} - \mu_{(k)}}{\mathrm{kl}(\mu_{(M)}, \mu_{(k)})} \log(T) + \mathbb{E}[T_0] M + o(\log(T)).$$

Thanks to Lemmas 1 and 2, $\mathbb{E}[T_0] = \mathcal{O}\left(\frac{K^3 \log(T)}{\mu_{(K)}}\right)$, yielding Theorem 1. ■

### A.2.2. PROOF OF LEMMA 1

Let $j$ be a cooperative player and $q_k(t)$ be the probability at round $t$ that the remaining player pulls $k$. Define $p_k^j(t) = \mathbb{P}[t \in C_k^j(t) \mid t \in N_k^j(t)]$. By definition, $p_k^j(t) = 1 - (1 - 1/K)^{M-2}(1 - q_k(t))$ when all cooperative players are pulling uniformly at random. Two auxiliary Lemmas using classical concentration inequalities are used to prove Lemma 1. The proofs of Lemmas 4 and 5 are given in Appendix A.2.6.

**Lemma 4.** *For any $\delta > 0$,*

1. $\mathbb{P}\left[\left| \frac{\# C_k^j(T_M)}{\# N_k^j(T_M)} - \frac{1}{\# N_k^j(T_M)} \sum_{t \in N_k^j(T_M)} p_k^j(t) \right| \geq \delta \;\middle|\; N_k^j(T_M) \right] \leq 2 \exp(-\frac{\# N_k^j(T_M) \delta^2}{2})$.

*For any $\delta \in (0, 1)$ and fixed $T_M$,*

2. $\mathbb{P}\left[\left|\#N_k^j - \frac{\alpha_k T_M}{K}\right| \geq \delta \frac{\alpha_k T_M}{K}\right] \leq 2\exp(-\frac{T_M \alpha_k \delta^2}{3K})$.

3. $\mathbb{P}\left[\left|\sum_{t=1}^{T_M}(\mathbb{1}(t \in N_k^j) - \frac{\alpha_k}{K})p_k^j(t)\right| \geq \delta \frac{\alpha_k T_M}{K}\right] \leq 2\exp\left(-\frac{T_M \alpha_k \delta^2}{3K}\right)$.

**Lemma 5.** *For any $k$, $j$ and $\delta \in (0, \frac{\alpha_k}{K})$, with probability larger than $1 - \frac{6KM}{T}$,*

$$\left|\hat{p}_k^j(t_m^j) - \frac{1}{t_m^j}\sum_{t=1}^{t_m^j} p_k^j(t)\right| \leq 2\sqrt{\frac{6\log(T)}{n\left(1 - 2\sqrt{\frac{3}{2\beta^2}(1+\frac{3}{2\beta^2})}\right)}} + 2\sqrt{\frac{\log(T)}{n}}.$$

*And for $\beta \geq 39$:*

$$\frac{t_m^j \alpha_{(k)}}{K} \in \left[\frac{13}{14}n,\ \frac{16}{15}n\right].$$

Let $\varepsilon = 2\sqrt{\frac{6\log(T)}{n\left(1-2\sqrt{\frac{3}{2\beta^2}(1+\frac{3}{2\beta^2})}\right)}} + 2\sqrt{\frac{\log(T)}{n}}$ and $p_k^j = \frac{1}{t_m^j}\sum_{t=1}^{t_m^j} p_k^j(t)$ such that with probability at least $1 - \frac{6KM}{T}$, $\left|\hat{p}_k^j - p_k^j\right| \leq \varepsilon$. The remaining of the proof is conditioned on this event.

By definition of $n$, $\varepsilon = \frac{1}{K}f(\beta)$ where $f(x) = \frac{2}{x}\sqrt{\frac{6}{1-2\sqrt{\frac{3}{2x^2}(1+\frac{3}{2x^2})}}} + 2/x$. Note that $f(x) \leq \frac{1}{2e}$ for $x \geq 39$ and thus $\varepsilon \leq \frac{1}{2Ke}$ for the considered $\beta$.

The last point of Lemma 5 yields that $t_m^j \leq \frac{\gamma_2}{\gamma_1} t_m^{j'}$ for any pair $j, j'$. All the cooperative players are thus pulling uniformly at random until at least $t_m^j$, thanks to the additional waiting room. Then,

$$\frac{1}{K}\sum_k (1 - p_k^j(t)) \ = \ (1 - 1/K)^{M-2}(1 - \frac{1}{K}\sum_k q_k(t)) \ = \ (1 - 1/K)^{M-1}.$$

When summing over $k$, it follows:

$$\begin{aligned}
\frac{1}{K}\sum_k (1 - p_k^j) - \varepsilon \leq \frac{1}{K}\sum_k (1 - \hat{p}_k^j) &\leq \frac{1}{K}\sum_k (1 - p_k^j) + \varepsilon \\
(1 - 1/K)^{M-1} - \varepsilon \leq \frac{1}{K}\sum_k (1 - \hat{p}_k^j) &\leq (1 - 1/K)^{M-1} + \varepsilon \\
M - 1 + \frac{\log(1 + \frac{\varepsilon}{(1-1/K)^{M-1}})}{\log(1 - 1/K)} \leq \frac{\log\left(\frac{1}{K}\sum_k (1 - \hat{p}_k^j)\right)}{\log(1 - 1/K)} &\leq M - 1 + \frac{\log(1 - \frac{\varepsilon}{(1-1/K)^{M-1}})}{\log(1 - 1/K)} \\
M - 1 + \frac{\log(1 + \frac{1}{2K})}{\log(1 - 1/K)} \leq \frac{\log\left(\frac{1}{K}\sum_k (1 - \hat{p}_k^j)\right)}{\log(1 - 1/K)} &\leq M - 1 + \frac{\log(1 - \frac{1}{2K})}{\log(1 - 1/K)}
\end{aligned}$$

The last line is obtained by observing that $\frac{\varepsilon}{(1-1/K)^{M-1}}$ is smaller than $\frac{1}{2K}$.

Observing that $\max\left(\frac{\log(1-x/2)}{\log(1-x)}, -\frac{\log(1+x/2)}{\log(1-x)}\right) < 1/2$ for any $x > 0$, the last line implies:

$$1 + \frac{\log\left(\frac{1}{K}\sum_k (1 - \hat{p}_k^j)\right)}{\log(1 - 1/K)} \in (M - 1/2, M + 1/2).$$

When rounding this quantity to the closest integer, we thus obtain $M$, which yields the first part of Lemma 1. The second part is directly given by Lemma 5. ■

### A.2.3. PROOF OF LEMMA 2

The proof of Lemma 2 relies on two lemmas given below.

**Lemma 6.** *Conditionally on the success of the estimation phase, when a cooperative player $j$ proceeds to* `GetRank`*, all other cooperative players are either running* `GetRank` *or in a waiting room*[5]*, i.e., they are not proceeding to* `Alternate Exploration` *yet.*

**Proof.** Recall that $\gamma_1 = 13/14$ and $\gamma_2 = 16/15$. Conditionally on the success of the estimation phase, for any pair $(j, j')$, $\frac{\gamma_2}{\gamma_1} t_m^j \geq t_m^{j'}$. Let $t_r^j = \frac{t_m^j}{\gamma_1 K^2 \beta^2}$ be the duration time of `GetRank` for player $j$. For the same reason, $\frac{\gamma_2}{\gamma_1} t_r^j \geq t_r^{j'}$. Player $j$ ends `GetRank` at round $t^j = \frac{\gamma_2}{\gamma_1} t_m^j + t_r^j$ and the second waiting room at round $\frac{\gamma_2}{\gamma_1} t^j$.

As $\frac{\gamma_2}{\gamma_1} t^j \geq t^{j'}$, this yields that when a player ends `GetRank`, all other players are not running `Selfish-Robust MMAB` yet. Because $\frac{\gamma_2}{\gamma_1} t_m^j \geq t_m^{j'}$, when a player starts `GetRank`, all other players also have already ended `EstimateM`. This yields Lemma 6. ■

**Lemma 7.** *Conditionally on the success of the estimation phase, with probability larger than $1 - \frac{1}{T}$, cooperative player $j$ ends* `GetRank` *with a rank in $[M]$.*

**Proof.** Conditionally on the success of the estimation phase and thanks to Lemma 5, $t_r^j = \frac{t_m^j}{\gamma_1 K^2 \beta^2} \geq \frac{K \log(T)}{\alpha_{(K)}}$. Moreover, at any round of `GetRank`, the probability of observing $\eta_k(t) = 0$ is larger than $\frac{\alpha_{(K)}}{M}$. Indeed, the probability of observing $\eta_k(t)$ is larger than $\alpha_{(K)}$ with Statistic sensing. Independently, the probability of having $\eta_k = 0$ is larger than $1/M$ since there is at least an arm among $[M]$ not pulled by any other player. These two points yield, as $M \leq K$:

$$\begin{aligned}
\mathbb{P}[\text{player does not observe } \eta_k(t) = 0 \text{ for } t_r^j \text{ successive rounds}] &\leq \left(1 - \frac{\alpha_{(K)}}{M}\right)^{t_r^j} \\
&\leq \exp\left(-\frac{\alpha_{(K)} t_r^j}{M}\right) \\
&\leq \frac{1}{T}
\end{aligned}$$

Thus, with probability larger than $1 - \frac{1}{T}$, player $j$ observes $\eta_k(t) = 0$ at least once during `GetRank`, i.e., she ends the procedure with a rank in $[M]$. ■

**Proof of Lemma 2.** Combining Lemmas 6 and 7 yields that the cooperative player $j$ ends `GetRank` with a rank in $[M]$ and no other cooperative player ends with the same rank. Indeed, when a player gets the rank $j$, any other cooperative player has either no attributed rank (still running `GetRank` or the first waiting room), or an attributed rank $j'$. In the latter case, thanks to Lemma 6, this other

5. Note that there is a waiting room before **and** after `GetRank`.

player is either running `GetRank` or in the second waiting room, meaning she is still pulling $j'$. Since the first player ends with the rank $j$, this means that she did not encounter a collision when pulling $j$ and especially, $j \neq j'$.

Considering a union bound among all cooperative players now yields Lemma 2. ■

#### A.2.4. PROOF OF LEMMA 3

Let us denote $T_0^j = \left(\frac{\gamma_2}{\gamma_1^2 \beta^2 K^2} + \frac{\gamma_2^2}{\gamma_1^2}\right) t_m^j$ such that player $j$ starts running `Alternate Exploration` at time $T_0^j$. This section aims at proving Lemma 3. In this section, the initialization is assumed to be successful. The regret due to an unsuccessful initialization is constant in $T$ and thus $o(\log(T))$. We prove in this section, in case of a successful initialization, the following:

$$\mathbb{E}[R^{\text{explo}}] \leq M \sum_{k>M} \frac{\mu_{(M)} - \mu_{(k)}}{\text{kl}(\mu_{(M)}, \mu_{(k)})} \log(T) + o(\log(T)). \tag{2}$$

This proof follows the same scheme as the regret proof from Proutiere and Wang (2019), except that there is no leader here. Every *bad event* then happens independently for each individual player. This adds a $M$ factor in the regret compared to the follower/leader algorithm[6] used by Proutiere and Wang (2019). For conciseness, we only give the main steps and refer to the original Lemmas in (Proutiere and Wang, 2019) for their detailed proof.

We first recall useful concentration Lemmas which correspond to Lemmas 1 and 2 in (Proutiere and Wang, 2019). They are respectively simplified versions of Lemma 5 in (Combes et al., 2015) and Theorem 10 in (Garivier and Cappé, 2011).

**Lemma 8.** *Let $k \in [K]$, $c > 0$ and $H$ be a (random) set such that for all $t$, $\{t \in H\}$ is $\mathcal{F}_{t-1}$ measurable. Assume that there exists a sequence $(Z_t)_{t\geq 0}$ of binary random variables, independent of all $\mathcal{F}_t$, such that for $t \in H$, $\pi^j(t) = k$ if $Z_t = 1$. Furthermore, if $\mathbb{E}[Z_t] \geq c$ for any $t$, then:*

$$\sum_{t\geq 1} \mathbb{P}[t \in H \mid |\widehat{\mu}_k^j(t) - \mu_k| \geq \delta] \leq \frac{4 + 2c/\delta^2}{c^2}.$$

**Lemma 9.** *If player $j$ starts following `Alternate Exploration` at round $T_0^j + 1$:*

$$\sum_{t > T_0^j} \mathbb{P}[b_k^j(t) < \mu_k] \leq 15.$$

Let $0 < \delta < \delta_0 := \min_k \frac{\mu_{(k)} - \mu_{(k+1)}}{2}$. Besides the definitions given in Appendix A.1, define the following:

- $\mathcal{M}^*$ the list of the $M$-best arms, ordered according to their indices.
- $\mathcal{A}^j = \{t > T_0^j \mid \mathcal{M}^j(t) \neq \mathcal{M}^*\}$.
- $\mathcal{D}^j = \{t > T_0^j \mid \exists k \in \mathcal{M}^j(t),\ |\widehat{\mu}_k^j(t) - \mu_k| \geq \delta\}$.

6. Which is not selfish-robust.

- $\mathcal{E}^j = \{t > T_0^j \mid \exists k \in \mathcal{M}^*,\ b_k^j(t) < \mu_k\}$.
- $\mathcal{G}^j = \{t \in \mathcal{A}^j \setminus \mathcal{D}^j \mid \exists k \in \mathcal{M}^* \setminus \mathcal{M}^j(t),\ |\widehat{\mu}_k^j(t) - \mu_k| \geq \delta\}$.

**Lemma 10.** $\mathbb{E}[\#(\mathcal{A}^j \cup \mathcal{D}^j)] \leq 8MK^2(6K + \delta^{-2})$.

**Proof.** Similarly to Proutiere and Wang (2019), we have $(\mathcal{A}^j \cup \mathcal{D}^j) \subset (\mathcal{D}^j \cup \mathcal{E}^j \cup \mathcal{G}^j)$. We can then individually bound $\mathbb{E}[\#\mathcal{D}^j]$, $\mathbb{E}[\#\mathcal{E}^j]$ and $\mathbb{E}[\#\mathcal{G}^j]$, leading to Lemma 10. The detailed proof is omitted here as it exactly corresponds to Lemmas 3 and 4 in (Proutiere and Wang, 2019). ∎

**Lemma 11.** *Consider a suboptimal arm $k$ and define $\mathcal{H}_k^j = \{t \in \{T_0^j + 1, \dots, T\} \setminus (\mathcal{A}^j \cup \mathcal{D}^j) \mid \pi^j(t) = k\}$. It holds*

$$\mathbb{E}\left[\#\mathcal{H}_k^j\right] \leq \frac{\log T + 4\log(\log T)}{\mathrm{kl}(\mu_k + \delta, \mu_{(M)} - \delta)} + 4 + 2\delta^{-2}.$$

Lemma 11 can be proved using the arguments of Lemma 5 in (Proutiere and Wang, 2019).

**Proof of Lemma 3.** If $t \in \mathcal{A}^j \cup \mathcal{D}^j$, player $j$ collides with at most one player $j'$ such that $t \notin \mathcal{A}^{j'} \cup \mathcal{D}^{j'}$.

Otherwise, $t \notin \mathcal{A}^j \cup \mathcal{D}^j$ and player $j$ collides with a player $j'$ only if $t \in \mathcal{A}^{j'} \cup \mathcal{D}^{j'}$. Also, she pulls a suboptimal arm $k$ only on an exploration slot, i.e., instead of pulling the $M$-th best arm. Thus, the regret caused by pulling a suboptimal arm $k$ when $t \notin \mathcal{A}^j \cup \mathcal{D}^j$ is $(\mu_{(M)} - \mu_k)$ and this actually happens when $t \in \mathcal{H}_k^j$.

This discussion provides the following inequality, which concludes the proof of Lemma 3 when using Lemmas 10 and 11 and taking $\delta \to 0$.

$$\mathbb{E}\left[R^{\text{explo}}\right] \leq \underbrace{2\sum_{j=1}^{M}\mathbb{E}\left[\#(\mathcal{A}^j \cup \mathcal{D}^j)\right]}_{\text{collisions}} + \underbrace{\sum_{j\leq M}\sum_{k>M}(\mu_{(M)} - \mu_{(k)})\mathbb{E}\left[\#\mathcal{H}_k^j\right]}_{\text{pulls of suboptimal arms}}.$$

∎

### A.2.5. PROOF OF THEOREM 2

1. Let us first prove the Nash equilibrium property. Define $\mathcal{E} = [T_0] \cup \left(\bigcup_{j\in[M]}(\mathcal{A}^j \cup \mathcal{D}^j)\right)$ with the definitions of $T_0, \mathcal{A}^j$ and $\mathcal{D}^j$ given in Appendix A.2.4. Thanks to Lemmas 1 and 2, regardless of the strategy of a selfish player, all other players successfully end the initialization after a time $T_0$ with probability $1 - \mathcal{O}(KM/T)$. The remaining of the proof is conditioned on this event.

The selfish player earns at most $\mu_{(1)}T_0$ during the initialization. Note that `Alternate Exploration` never uses collision information, meaning that the behavior of the strategic player during this phase does not change the behaviors of the cooperative players. Thus, the optimal strategy during this phase for the strategic player is to pull the best available arm. Let $j$ be the rank of the strategic player[7]. For $t \notin \mathcal{E}$, this arm is the $k$-th arm of $\mathcal{M}^*$ with $k = t + j \pmod M + 1$. In a whole block of length $M$ in $[T] \setminus \mathcal{E}$, the selfish player then earns at most $\sum_{k=1}^M \mu_{(k)}$.

Over all, when a strategic player deviates from `Alternate Exploration`, she earns at most:

$$\mathbb{E}[\mathrm{Rew}_T^j(s', s_{-j})] \leq \mu_{(1)}(\#\mathcal{E} + M) + \frac{T}{M}\sum_{k=1}^M \mu_{(k)}.$$

Note that we here add a factor $\mu_{(1)}$ in the initialization regret. This is only because the true loss of colliding is not 1 but $\mu_{(1)}$. Also, the additional $\mu_{(1)}M$ term is due to the fact that the last block of length $M$ of `Alternate Exploration` is not totally completed.

Thanks to Theorem 1, it also comes:

$$\mathbb{E}[\mathrm{Rew}_T^j(s)] \geq \frac{T}{M}\sum_{k=1}^M \mu_{(k)} - \sum_{k>M} \frac{\mu_{(M)} - \mu_{(k)}}{\mathrm{kl}(\mu_{(k)}, \mu_{(M)})}\log(T) - \mathcal{O}\left(\mu_{(1)}\frac{K^3}{\mu_{(K)}}\log(T)\right).$$

Lemmas 2 and 10 yield that $\mathbb{E}[\#\mathcal{E}] = \mathcal{O}\left(\frac{K^3\log(T)}{\mu_{(K)}}\right)$, which concludes the proof.

2. We now prove the $(\alpha, \varepsilon)$-stability of `Selfish-Robust MMAB`. Let $\varepsilon' = \mathbb{E}[\mathcal{E}] + M$. Consider that player $j$ is playing a deviation strategy $s' \in \mathcal{S}$ such that for some other player $i$ and $l > 0$:

$$\mathbb{E}[\mathrm{Rew}_T^i(s', s_{-j})] \leq \mathbb{E}[\mathrm{Rew}_T^i(s)] - l - (\varepsilon' + M).$$

We will first compare the reward of player $j$ with her optimal possible reward. The only way for the selfish player to influence the sampling strategy of another player is in modifying the rank attributed to this other player. The total rewards of cooperative players with ranks $j$ and $j'$ only differ by at most $\varepsilon' + M$ in expectation, without considering the loss due to collisions with the selfish player.

The only other way to cause regret to another player $i$ is then to pull $\pi^i(t)$ at time $t$. This incurs a loss at most $\mu_{(1)}$ for player $i$, while this incurs a loss at least $\mu_{(M)}$ for player $j$, in comparison with her optimal strategy. This means that for incurring the additional loss $l$ to the player $i$, player $j$ must suffer herself from a loss $\frac{\mu_{(M)}}{\mu_{(1)}}$ compared to her optimal strategy $s^*$. Thus, for $\alpha = \frac{\mu_{(M)}}{\mu_{(1)}}$:

$$\mathbb{E}[\mathrm{Rew}_T^i(s', s_{-j})] \leq \mathbb{E}[\mathrm{Rew}_T^i(s)] - l - (\varepsilon' + M) \implies \mathbb{E}[\mathrm{Rew}_T^j(s', s_{-j})] \leq \mathbb{E}[\mathrm{Rew}_T^j(s^*, s_{-j})] - \alpha l$$

The first point of Theorem 2 yields for its given $\varepsilon$: $\mathbb{E}[\mathrm{Rew}_T^j(s^*, s_{-j})] \leq \mathbb{E}[\mathrm{Rew}_T^j(s)] + \varepsilon$.

Noting $l_1 = l + \varepsilon' + M$ and $\varepsilon_1 = \varepsilon + \alpha(\varepsilon' + M) = \mathcal{O}(\varepsilon)$, we have shown:

$$\mathbb{E}[\mathrm{Rew}_T^i(s', s_{-j})] \leq \mathbb{E}[\mathrm{Rew}_T^i(s)] - l_1 \implies \mathbb{E}[\mathrm{Rew}_T^j(s', s_{-j})] \leq \mathbb{E}[\mathrm{Rew}_T^j(s)] + \varepsilon_1 - \alpha l_1.$$

■

7. If the strategic player has no attributed rank, it is the only non-attributed rank in $[M]$.

### A.2.6. AUXILIARY LEMMAS

This section provides useful Lemmas for the proof of Lemma 1. We first recall a useful version of Chernoff bound.

**Lemma 12.** *For any independent variables $X_1, \ldots, X_n$ in $[0,1]$ and $\delta \in (0,1)$:*

$$\mathbb{P}\left(\left|\sum_{i=1}^{n} X_i - \mathbb{E}[X_i]\right| \geq \delta \sum_{i=1}^{n} \mathbb{E}[X_i]\right) \leq 2e^{-\frac{\delta^2 \sum_{i=1}^{n} \mathbb{E}[X_i]}{3}}.$$

**Proof of Lemma 4.**

1. This is an application of Azuma-Hoeffding inequality on the variables $\mathbb{1}(t \in C_k^j(T_M)) \mid t \in N_k^j(T_M)$.

2. This is a consequence of Lemma 12 on the variables $\mathbb{1}(t \in N_k^j)$.

3. This is the same result on the variables $\mathbb{1}(t \in N_k^j)p_k^j(t) \,|\mathcal{F}_{t-1}$ where $\mathcal{F}_{t-1}$ is the filtration associated to the past events, using $\sum_{t=1}^{T_M} \mathbb{E}[\mathbb{1}(t \in N_k^j)p_k^j(t)|\mathcal{F}_{t-1}] \leq \frac{T_M \alpha_k}{K}$.

■

**Proof of Lemma 5.** From Lemma 4, it comes:

- $\mathbb{P}\left[\exists t \leq T, \left|\hat{p}_k^j(t) - \frac{1}{\#N_k^j} \sum_{t' \in N_k^j} p_k^j(t')\right| \geq 2\sqrt{\frac{\log(T)}{\#N_k^j}}\right] \leq \frac{2}{T}$,

- $\mathbb{P}\left[\exists t \leq T, \left|\frac{K \#N_k^j}{\alpha_k t} - 1\right| \geq \sqrt{\frac{6\log(T)K}{\alpha_k t}}\right] \leq \frac{2}{T}$, (3)

- $\mathbb{P}\left[\exists t \leq T, \left|\frac{K}{\alpha_k t} \sum_{t' \in N_k^j} p_k^j(t') - \frac{1}{t}\sum_{t' \leq t} p_k^j(t')\right| \geq \sqrt{\frac{6\log(T)K}{\alpha_k t}}\right] \leq \frac{2}{T}$.

Noting that $\sum_{t' \in N_k^j} p_k^j(t') \leq \#N_k^j$, Equation (3) implies:

$$\mathbb{P}\left[\exists t \leq T, \left|\frac{K}{\alpha_k t} \sum_{t' \in N_k^j} p_k^j(t') - \frac{1}{\#N_k^j} \sum_{t' \in N_k^j} p_k^j(t')\right| \geq \sqrt{\frac{6\log(T)K}{\alpha_k t}}\right] \leq \frac{2}{T}.$$

Combining these three inequalities and making the union bound over all the players and arms yield that with probability larger than $1 - \frac{6KM}{T}$:

$$\left|\hat{p}_k^j(t_m^j) - \frac{1}{t_m^j} \sum_{t \leq t_m^j} p_k^j(t)\right| \leq 2\sqrt{\frac{6\log(T)K}{\alpha_k t_m^j}} + 2\sqrt{\frac{\log(T)}{\#N_k^j(t_m^j)}}. \tag{4}$$

Moreover, under the same event, Equation (3) also gives that

$$N_k^j(t_m^j) \in \left[\frac{\alpha_k t_m^j}{K} - \sqrt{\frac{6\alpha_k t_m^j \log(T)}{K}}, \frac{\alpha_k t_m^j}{K} + \sqrt{\frac{6\alpha_k t_m^j \log(T)}{K}}\right].$$

Specifically, this yields $n \leq \frac{\alpha_k t_m^j}{K} + \sqrt{\frac{6\alpha_k t_m^j \log(T)}{K}}$, or equivalently $\frac{t_m^j \alpha_k}{K} \geq n - 2\sqrt{\frac{3\log(T)}{2}}\sqrt{n + \frac{3\log(T)}{2}}$. Since $n = \beta^2 K^2 \log(T)$, this becomes $\frac{t_m^j \alpha_k}{K} \geq n(1 - 2\sqrt{\frac{3}{2\beta^2 K^2}}\sqrt{1 + \frac{3}{2\beta^2 K^2}})$ and Equation (4) now rewrites into:

$$\left|\hat{p}_k^j(t_m^j) - \frac{1}{t_m^j}\sum_{t \leq t_m^j} p_k^j(t)\right| \leq 2\sqrt{\frac{6\log(T)}{n\left(1 - 2\sqrt{\frac{3}{2\beta^2K^2}(1 + \frac{3}{2\beta^2K^2})}\right)}} + 2\sqrt{\frac{\log(T)}{n}}$$

Also, $n \geq \frac{\alpha_k t_m^j}{K} - \sqrt{\frac{6\log(T)\alpha_k t_m^j}{K}}$ for some $k$, which yields $\frac{t_m^j \alpha_k}{K} \leq n(1 + \frac{3}{\beta^2 K^2} + 2\sqrt{\frac{3}{2\beta^2 K^2}}\sqrt{1 + \frac{3}{2\beta^2 K^2}})$. This relation then also holds for $\frac{t_m^j \alpha_{(K)}}{K}$. We have therefore proved that:

$$n\left(1 - 2\sqrt{\frac{3}{2\beta^2}}\sqrt{1 + \frac{3}{2\beta^2}}\right) \leq \frac{t_m^j \alpha_{(k)}}{K} \leq n\left(1 + \frac{3}{\beta^2} + 2\sqrt{\frac{3}{2\beta^2}}\sqrt{1 + \frac{3}{2\beta^2}}\right).$$

For $\beta \geq 39$, this gives the bound in Lemma 5. ∎

## Appendix B. Collective punishment proof

Recall that the punishment protocol consists in pulling each arm $k$ with probability at least $p_k^j = \max\left(1 - \left(\gamma \frac{\sum_{l=1}^M \widehat{\mu}_{(l)}^j}{M\widehat{\mu}_k^j}\right)^{\frac{1}{M-1}}, 0\right)$. Lemma 13 below guarantees that such a sampling strategy is possible.

**Lemma 13.** *For $p_k = \max\left(1 - \left(\frac{\gamma \sum_{l=1}^M \widehat{\mu}_{(l)}^j}{M\widehat{\mu}_k^j}\right)^{\frac{1}{M-1}}, 0\right)$ with $\gamma = (1 - 1/K)^{M-1}$: $\sum_{k=1}^K p_k \leq 1$.*

**Proof.** For ease of notation, define $x_k := \widehat{\mu}_k^j$, $\bar{x}_M := \sum_{l=1}^M x_{(l)}/M$ and $S := \{k \in [K] \mid x_k > \gamma\bar{x}_M\} = \{k \in [K] \mid p_k > 0\}$. We then get by concavity of $x \mapsto -x^{-\frac{1}{M-1}}$,

$$\sum_{k \in S} p_k = \#S \times \left(1 - (\gamma\bar{x}_M)^{\frac{1}{M-1}} \sum_{k \in S} \frac{(x_k)^{-\frac{1}{M-1}}}{\#S}\right), \tag{5}$$

$$\leq \#S \times \left(1 - \left(\frac{\gamma\bar{x}_M}{\bar{x}_S}\right)^{\frac{1}{M-1}}\right) \qquad \text{with } \bar{x}_S = \frac{1}{\#S}\sum_{k \in S} x_k. \tag{6}$$

We distinguish two cases.

First, if $\#S \leq M$, we then get $M\bar{x}_M \geq \#S\bar{x}_S$ because $S$ is a subset of the $M$ best empirical arms. The last inequality then becomes

$$\sum_{k \in S} p_k \leq \#S\left(1 - \left(\gamma\frac{\#S}{M}\right)^{\frac{1}{M-1}}\right).$$

Define $g(x) = \frac{\gamma}{M} - x(1-x)^{M-1}$. For $x \in (0, 1]$:

$$\begin{aligned} g(x) \geq 0 &\iff \frac{\gamma}{xM} \geq (1-x)^{M-1}, \\ &\iff 1 - \left(\frac{\gamma}{xM}\right)^{\frac{1}{M-1}} \leq x, \\ &\iff \frac{1}{x}\left(1 - \left(\frac{\gamma}{xM}\right)^{\frac{1}{M-1}}\right) \leq 1. \end{aligned}$$

Thus, $g(\frac{1}{\#S}) \geq 0$ implies $\sum_{k\in S} p_k \leq 1$. We now show that $g$ is indeed non negative on $[0, 1]$. $x(1-x)^{M-1}$ is maximized at $1/M$ and is thus smaller than $\frac{1}{M}(1-1/M)^{M-1}$, and using the fact that $\frac{1}{M}(1-1/M)^{M-1} \leq \frac{\gamma}{M}$ for our choice of $\gamma$, we get the result for the first case.

The other case corresponds to $\#S > M$. In this case, the $M$ best empirical arms are all in $S$ and thus $\bar{x}_M \geq \bar{x}_S$. Equation (6) becomes:

$$\sum_{k\in S} p_k \leq \#S\left(1 - \gamma^{\frac{1}{M-1}}\right) \leq K(1-(1-1/K)) = 1.$$

■

## Appendix C. Supplementary material for SIC-GT

In this whole section, $M$ is assumed to be at least 3.

### C.1. Description of the algorithm

This section provides a complete description of SIC-GT. The pseudocode of SIC-GT is given in Algorithm 3 and relies on several auxiliary protocols, which are described by Protocols 3, 4, 5, 6, 7, 8 and 9.

**Protocol 5:** `ReceiveMean`

```
Input: j, p
μ̃ ← 0
for n = 0, ..., p do
    Pull j
    if η_j(t) = 1 then μ̃ ← μ̃ + 2^{-n}
end
return μ̃                // sent mean
```

**Protocol 6:** `SendMean`

```
Input: j, l, p, μ̃
m ← dyadic writing of μ̃ of length p + 1,
  i.e., μ̃ = Σ_{n=0}^{p} m_n 2^{-n}
for n = 0, ..., p do
    if m_n = 1 then Pull l   // send 1
    else Pull j              // send 0
end
```

**Initialization phase.** The purpose of the initialization phase is to estimate $M$ and attribute ranks in $[M]$ to all the players. This is done by `Initialize`, which is given in Protocol 3. It simply consists in pulling uniformly at random for a long time to infer $M$ from the probability of collision. Then it proceeds to a Musical Chairs procedure so that each player ends with a different arm in $[M]$, corresponding to her rank.

**Algorithm 3:** `SIC-GT`

```
Input: T, δ
M, j ← Initialize (T, K) and punish ← False
OptArms ← ∅, M_p ← M, [K_p] ← [K] and p ← 1
while not punish and #OptArms < M do
    for m = 0, ..., ⌈K_p 2^p / M_p⌉ − 1 do
        ArmstoPull ← OptArms ∪ {i ∈ [K_p] | i − mM_p (mod K_p) ∈ [M_p]}
        for M rounds do
            k ← j + t (mod M) + 1 and pull i the k-th element of ArmstoPull
            if T_i^j(p) ≤ 2^p then Update μ̂_i^j        // T_i^j pulls on i by j this phase
            if η_i = 1 then punish ← True              // collisionless exploration
        end
    end
    (punish, OptArms, [K_p], M_p) ← CommPhase (μ̂^j, j, p, OptArms, [K_p], M_p)
    p ← p + 1
end
if punish then PunishHomogeneous (p)
else                                                   // exploitation phase
    k ← j + t (mod M) + 1 and pull i, the k-th arm of OptArms
    if η_i = 1 then punish ← True
end
```

**Protocol 3:** `Initialize`

```
Input: T, K
n_coll ← 0 and j ← −1
for 12eK^2 log(T) rounds do Pull k ~ U(K) and n_coll ← n_coll + η_k     // estim. M
M̂ ← 1 + round( log(1 − n_coll / (12eK^2 log(T))) / log(1 − 1/K) )
for K log(T) rounds do                                                // get rank
    if j = −1 then
        Pull k ~ U(M̂);   if η_k = 0 then j ← k
    else Pull j
end
return (M̂, j)
```

**Exploration phase.** As explained in Section 5.2, each arm that still needs to be explored (those in $[K_p]$, with Algorithm 3 notations) is pulled at least $M2^p$ times during the $p$-th exploration phase. Moreover, as soon as an arm is found optimal, it is pulled for each remaining round of the exploration. The last point is that each arm is pulled the exact same amount of time by any player, in order to ensure fairness of the algorithm, while still avoiding collisions. This is the interest of the ArmstoPull set in Algorithm 3. At each time step, the pulled arms are the optimal ones and $M_p$ arms that still

**Protocol 4:** `CommPhase`

**Input:** $\widehat{\mu}^j, j, p$, OptArms, $[K_p], M_p$
punish $\leftarrow$ False
**for** $K$ *rounds* **do** // receive punishment signal
  Pull $k = t + j \pmod K + 1$; **if** $\eta_k = 1$ **then** punish $\leftarrow$ True
**end**
$$\widetilde{\mu}_k^j \leftarrow \begin{cases} 2^{-p}\left(\lfloor 2^p \widehat{\mu}_k^j \rfloor + 1\right) \text{ with proba } 2^p\widehat{\mu}_k^j - \lfloor 2^p \widehat{\mu}_k^j \rfloor \\ 2^{-p}\lfloor 2^p \widehat{\mu}_k^j \rfloor \text{ otherwise} \end{cases}$$ // quantization
**for** $(i, l, k) \in [M] \times \{1, 2\} \times [K]$ *such that* $i \neq l$ **do** // $i$ sends $\widetilde{\mu}_k^i$ to $l$
  **if** $j = i$ **then** // sending player
    `SendMean` $(j, l, p, \widetilde{\mu}_k^j)$ and $q \leftarrow$ `ReceiveMean` $(j, p)$ // back and forth
    **if** $q \neq \widetilde{\mu}_k^j$ **then** punish $\leftarrow$ True // corrupted message
  **else if** $j = l$ **then** $\widetilde{\mu}_k^i \leftarrow$ `ReceiveMean` $(j, p)$ and `SendMean` $(j, i, p, \widetilde{\mu}_k^i)$
  **else** Pull $j$ // waiting for others
**end**
**for** $(i, l, m, k) \in \{(1, 2), (2, 1)\} \times [M] \times [K]$ **do** // leaders check info match
  **if** $j = i$ **then** `SendMean` $(j, l, p, \widetilde{\mu}_k^m)$
  **else if** $j = l$ **then**
    $q \leftarrow$ `ReceiveMean` $(j, p)$; **if** $q \neq \widetilde{\mu}_k^m$ **then** punish $\leftarrow$ True // info differ
  **else** Pull $j$ // waiting for leaders
**end**
**if** $j \in \{1, 2\}$ **then** (Acc, Rej) $\leftarrow$ `RobustUpdate` $(\widetilde{\mu}, p$, OptArms , $[K_p], M_p)$
**else** Acc, Rej $\leftarrow \emptyset$ // arms to accept/reject
(punish, Acc) $\leftarrow$ `SignalSet` (Acc, $j$, punish)
(punish, Rej) $\leftarrow$ `SignalSet` (Rej, $j$, punish)
**return** (punish, OptArms $\cup$ Acc, $[K_p] \setminus (\text{Acc} \cup \text{Rej})$ , $M_p - \#\text{Acc}$)

**Protocol 7:** `RobustUpdate`

**Input:** $\widetilde{\mu}, p$, OptArms , $[K_p], M_p$
Define for all $k$, $i^k \leftarrow \arg\max_{j \in [M]} \widetilde{\mu}_k^j$ and $i_k \leftarrow \arg\min_{j \in [M]} \widetilde{\mu}_k^j$
$\widetilde{\mu}_k \leftarrow \sum_{j \in [M] \setminus \{i^k, i_k\}} \widetilde{\mu}_k^j$ and $b \leftarrow 4\sqrt{\frac{\log(T)}{(M-2)2^{p+1}}}$
Rej $\leftarrow$ set of arms $k$ verifying $\#\{i \in [K_p] \,|\, \widetilde{\mu}_i - b \geq \widetilde{\mu}_k + b\} \geq M_p$
Acc $\leftarrow$ set of arms $k$ verifying $\#\{i \in [K_p] \,|\, \widetilde{\mu}_k - b \geq \widetilde{\mu}_i + b\} \geq K_p - M_p$
**return** (Acc, Rej)

need to be explored. The players proceed to a sliding window over these arms to explore, so that the difference in pulls for two arms in $[K_p]$ is at most 1 for any player and phase.

**Communication phase.** The pseudocode for a whole communication phase is given by `CommPhase` in Protocol 4. Players first quantize their empirical means before sending them in $p$ bits to each leader. The protocol to send a message is given by Protocol 6, while Protocol 5 describes how to receive the message. The messages are sent using back and forth procedures to detect corrupted messages.

**Protocol 8:** `SignalSet`

**Input:** $S, j,$ punish
length_S $\leftarrow \#S$ `// length of` $S$ `for leaders, 0 for others`
**for** $K$ *rounds* **do** `// leaders send` $\#S$
  **if** $j \in \{1, 2\}$ **then** Pull length_S
  **else**
    Pull $k = t + j \pmod K + 1$
    **if** $\eta_k = 1$ *and length_S* $\neq 0$ **then** punish $\leftarrow$ True `// receive different info`
    **if** $\eta_k = 1$ *and length_S* $= 0$ **then** length_S $\leftarrow k$
**end**
**for** $n = 1, \ldots,$ length_S **do** `// send/receive` $S$
  **for** $K$ *rounds* **do**
    **if** $j \in \{1, 2\}$ **then** Pull $n$-th arm of $S$
    **else**
      Pull $k = t + j \pmod K + 1$;  **if** $\eta_k = 1$ **then** Add $k$ to S
  **end**
**end**
**if** $\#S \neq$ length_S **then** punish $\leftarrow$ True `// corrupted info`
**return** (punish, $S$)

**Protocol 9:** `PunishHomogeneous`

**Input:** $p$
**if** *communication phase* $p$ *starts in less than* $M$ *rounds* **then**
  **for** $M + K$ *rounds* **do** Pull $j$ `// signal punish to everyone`
**else for** $M$ *rounds* **do** Pull the first arm of ArmstoPull as defined in Algorithm 3

$\gamma \leftarrow (1 - 1/K)^{M-1}$ and $\delta = \frac{1-\gamma}{1+3\gamma}$;  Set $\widehat{\mu}_k^j, S_k^j, s_k^j, n_k^j \leftarrow 0$
**while** $\exists k \in [K], \delta\widehat{\mu}_k^j < 2s_k^j(\log(T)/n_k^j)^{1/2} + \frac{14\log(T)}{3(n_k^j-1)}$ **do** `// estimate` $\mu_k$
  Pull $k = t + j \pmod K + 1$
  **if** $\delta\widehat{\mu}_k^j < 2s_k^j(\log(T)/n_k^j)^{1/2} + \frac{14\log(T)}{3(n_k^j-1)}$ **then**
    Update $\widehat{\mu}_k^j \leftarrow \frac{n_k^j}{n_k^j+1}\widehat{\mu}_k^j + X_k(t)$ and $n_k^j \leftarrow n_k^j + 1$
    Update $S_k^j \leftarrow S_k^j + X_k^2$ and $s_k^j \leftarrow \sqrt{\frac{S_k^j - (\widehat{\mu}_k^j)^2}{n_k^j - 1}}$
**end**
$p_k \leftarrow \left(1 - \left(\gamma\frac{\sum_{l=1}^M \widehat{\mu}_{(l)}^j(t)}{M\widehat{\mu}_k^j(t)}\right)^{\frac{1}{M-1}}\right)_+$;  $\widetilde{p}_k \leftarrow p_k / \sum_{l=1}^K p_l$ `// renormalize`
**while** $t \leq T$ **do** Pull $k$ with probability $p_k$ `// punish`

After this, leaders communicate the received statistics to each other, to ensure that no player sent differing ones to them.

They can then determine which arms are optimal/suboptimal using `RobustUpdate` given by Protocol 7. As explained in Section 5.2, it cuts out the extreme estimates and decides based on the $M-2$ remaining ones.

Afterwards, the leaders signal to the remaining players the sets of optimal and suboptimal arms as described by Protocol 8. If the leaders send differing information, it is detected by at least one player.

If the presence of a malicious player is detected at some point of this communication phase, then players signal to each other to trigger the punishment protocol described by Protocol 9.

**Exploitation phase.** If no malicious player perturbed the communication, players end up having detected the $M$ optimal arms. As soon as it is the case, they only pull these $M$ arms in a collisionless way until the end.

### C.2. Regret analysis

This section aims at proving the first point of Theorem 6, using similar techniques as in (Boursier and Perchet, 2019). The regret is first divided into three parts:

$$R_T = R^{\text{init}} + R^{\text{comm}} + R^{\text{explo}}, \tag{7}$$

$$\text{where } \begin{cases} R^{\text{init}} = T_{\text{init}} \sum_{k=1}^{M} \mu_{(k)} - \mathbb{E}_\mu \Big[ \sum_{t=1}^{T_{\text{init}}} \sum_{j=1}^{M} r^j(t) \Big] \text{ with } T_{\text{init}} = (12eK^2 + K)\log(T), \\ R^{\text{comm}} = \mathbb{E}_\mu \Big[ \sum_{t \in \text{Comm}} \sum_{j=1}^{M} (\mu_{(j)} - r^j(t)) \Big] \text{ with Comm the set of communication steps,} \\ R^{\text{explo}} = \mathbb{E}_\mu \Big[ \sum_{t \in \text{Explo}} \sum_{j=1}^{M} (\mu_{(j)} - r^j(t)) \Big] \text{ with Explo} = \{T_{\text{init}} + 1, \ldots, T\} \setminus \text{Comm}. \end{cases}$$

A communication step is defined as a round where any player is using the `CommPhase` protocol. Lemma 14 provides guarantees about the initialization phase. When all players correctly estimate $M$ and have different ranks after the protocol `Initialize`, the initialization phase is said successful.

**Lemma 14.** *Independently of the sampling strategy of the selfish player, if all other players follow* `Initialize`*, with probability at least* $1 - \frac{3M}{T}$*:* $\widehat{M}^j = M$ *and all cooperative players end with different ranks in* $[M]$*.*

**Proof.** Let $q_k(t) = \mathbb{P}[\text{selfish player pulls } k \text{ at time } t]$. Then, for any cooperative player $j$ during the initialization phase:

$$\begin{aligned} \mathbb{P}[\text{player } j \text{ observes a collision at time } t] &= \sum_{k=1}^{K} \frac{1}{K} (1 - 1/K)^{M-2} (1 - q_k(t)) \\ &= (1 - 1/K)^{M-2} (1 - \frac{\sum_{k=1}^{K} q_k(t)}{K}) \\ &= (1 - 1/K)^{M-1} \end{aligned}$$

Define $p = (1 - 1/K)^{M-1}$ the probability to collide and $\hat{p}^j = \frac{\sum_{t=1}^{12eK^2 \log(T)} \mathbb{1}_{\eta_{\pi^j(t)}=1}}{12eK^2 \log(T)}$ its estimation by player $j$. The Chernoff bound given by Lemma 12 gives:

$$\begin{aligned} \mathbb{P}\left[ \left| \hat{p}^j - p \right| \geq \frac{p}{2K} \right] &\leq 2e^{-\frac{p \log(T)}{e}} \\ &\leq 2/T \end{aligned}$$

If $\left| \hat{p}^j - p \right| < \frac{p}{2K}$, using the same reasoning as in the proof of Lemma 1 leads to $1 + \frac{\log(1-\hat{p}^j)}{\log(1-1/K)} \in (M - 1/2, M + 1/2)$ and then $\widehat{M}^j = M$. With probability at least $1 - 2M/T$, all cooperative players correctly estimate $M$.

Afterwards, the players sample uniformly in $[M]$ until observing no collision. As at least an arm in $[M]$ is not pulled by any other player, at each time step of this phase, when pulling uniformly at random:

$$\mathbb{P}[\eta_{\pi^j(t)} = 0] \geq 1/M.$$

A player gets a rank as soon as she observes no collision. With probability at least $1-(1-1/M)^n$, she thus gets a rank after at most $n$ pulls during this phase. Since this phase lasts $K\log(T)$ pulls, she ends the phase with a rank with probability at least $1-1/T$. Using a union bound finally yields that every player ends with a rank and a correct estimation of $M$. Moreover, these ranks are different between all the players, because a player fixes to the arm $j$ as soon as she gets attributed the rank $j$. ■

Lemma 15 bounds the exploration regret of `SIC-GT` and is proved in Appendix C.2.1. Note that a minimax bound can also be proved as done in (Boursier and Perchet, 2019).

**Lemma 15.** *If all players follow* `SIC-GT`*, with probability* $1-\mathcal{O}\left(\frac{KM\log(T)}{T}\right)$,

$$R^{explo} = \mathcal{O}\left(\sum_{k>M} \frac{\log(T)}{\mu_{(M)} - \mu_{(k)}}\right).$$

Lemma 16 finally bounds the communication regret.

**Lemma 16.** *If all players follow* `SIC-GT`*, with probability* $1-\mathcal{O}\left(\frac{KM\log(T)}{T}+\frac{M}{T}\right)$*:*

$$R^{comm} = \mathcal{O}\left(M^2K\log^2\left(\frac{\log(T)}{(\mu_{(M)} - \mu_{(M+1)})^2}\right)\right).$$

**Proof.** The proof is conditioned on the success of the initialization phase, which happens with probability $1-\mathcal{O}\left(\frac{M}{T}\right)$. Proposition 1 given in Appendix C.2.1 yields that with probability $1-\mathcal{O}\left(\frac{KM\log(T)}{T}\right)$, the number of communication phases is bounded by $N=\mathcal{O}\left(\log\left(\frac{\log(T)}{(\mu_{(M)}-\mu_{(M+1)})^2}\right)\right)$. The $p$-th communication phase lasts $8MK(p+1)+3K+K\,\#\text{Acc}(p)+K\,\#\text{Rej}(p)$, where Acc and Rej respectively are the accepted and rejected arms at the $p$-th phase. Their exact definitions are given in Protocol 7. An arm is either accepted or rejected only once, so that $\sum_{p=1}^N \#\text{Acc}(p)+\#\text{Rej}(p)=K$. The total length of Comm is thus bounded by:

$$\begin{aligned}\#\text{Comm} &\leq \sum_{p=1}^{N} 8MK(p+1)+3K+K\,\#\text{Acc}(p)+K\,\#\text{Rej}(p)\\ &\leq 8MK\frac{(N+2)(N+1)}{2}+3KN+K^2\end{aligned}$$

Which leads to $R^{\text{comm}}=\mathcal{O}\left(M^2K\log^2\left(\frac{\log(T)}{(\mu_{(M)}-\mu_{(M+1)})^2}\right)\right)$ using the given bound for $N$. ■

**Proof of Theorem 6.** Using Lemmas 14, 15, 16 and equation (7) it comes that with probability $1 - \mathcal{O}\left(\frac{KM\log(T)}{T}\right)$:

$$R_T \leq \mathcal{O}\left(\sum_{k>M} \frac{\log(T)}{\mu_{(M)} - \mu_{(k)}} + M^2 K \log^2\left(\frac{\log(T)}{(\mu_{(M)} - \mu_{(M+1)})^2}\right) + MK^2\log(T)\right).$$

The regret incurred by the low probability event is $\mathcal{O}(KM^2\log(T))$, leading to Theorem 6. ■

### C.2.1. PROOF OF LEMMA 15

Lemma 15 relies on the following concentration inequality.

**Lemma 17.** *Conditioned on the success of the initialization and independently of the means sent by the selfish player, if all other players play cooperatively and send uncorrupted messages, for any $k \in [K]$:*

$$\mathbb{P}[\exists p \leq n, |\widetilde{\mu}_k(p) - \mu_k| \geq B(p)] \leq \frac{4nM}{T}$$

*where $B(p) = 4\sqrt{\frac{\log(T)}{(M-2)2^{p+1}}}$ and $\widetilde{\mu}_k(p)$ is the centralized mean of arm $k$ at the end of phase $p$, once the extremes have been cut out. It exactly corresponds to the $\widetilde{\mu}_k$ of Protocol 7.*

**Proof.** At the end of phase $p$, $(2^{p+1}-1)$ observations are used for any player $j$ and arm $k$. Hoeffding bound then gives: $\mathbb{P}\left[\left|\widehat{\mu}_k^j(p) - \mu_k\right| \geq \sqrt{\frac{\log(T)}{2^{p+1}}}\right] \leq \frac{2}{T}$. The quantization only adds an error of at most $2^{-p}$, yielding for any cooperative player:

$$\mathbb{P}\left[\left|\widetilde{\mu}_k^j(p) - \mu_k\right| \geq 2\sqrt{\frac{\log(T)}{2^{p+1}}}\right] \leq \frac{2}{T} \tag{8}$$

Assume w.l.o.g. that the selfish player has rank $M$. Hoeffding inequality also yields:

$$\mathbb{P}\left[\left|\frac{1}{M-1}\sum_{j=1}^{M-1} \widehat{\mu}_k^j(p) - \mu_k\right| \geq \sqrt{\frac{\log(T)}{(M-1)2^{p+1}}}\right] \leq \frac{2}{T}.$$

Since $\sum_{j=1}^{M-1} 2^p(\widetilde{\mu}_k^j(p) - \widehat{\mu}_k^j(p))$ is the difference between $M-1$ Bernoulli variables and their expectation, Hoeffding inequality yields $\mathbb{P}\left[\left|\frac{1}{M-1}\sum_{j=1}^{M-1}(\widetilde{\mu}_k^j - \widehat{\mu}_k^j(p))\right| \geq \sqrt{\frac{\log(T)}{(M-1)2^{p+1}}}\right] \leq \frac{2}{T}$ and:

$$\mathbb{P}\left[\left|\frac{1}{M-1}\sum_{j=1}^{M-1} \widetilde{\mu}_k^j(p) - \mu_k\right| \geq 2\sqrt{\frac{\log(T)}{(M-1)2^{p+1}}}\right] \leq \frac{4}{T}. \tag{9}$$

Using the triangle inequality combining equations (8) and (9) yields for any $j \in [M-1]$:

$$
\begin{aligned}
\mathbb{P}\left[\left|\frac{1}{M-2}\sum_{\substack{j'\in[M-1]\\ j'\neq j}}\widetilde{\mu}_k^{j}(p)-\mu_k\right| \geq 4\sqrt{\frac{\log(T)}{(M-2)2^{p+1}}}\right] &\leq \mathbb{P}\left[\frac{M-1}{M-2}\left|\frac{1}{M-1}\sum_{j'\in[M-1]}\widetilde{\mu}_k^{j}(p)-\mu_k\right|\right. \\
&\qquad \left. + \frac{1}{M-2}\left|\widetilde{\mu}_k^{j}(p)-\mu_k\right| \geq 4\sqrt{\frac{\log(T)}{(M-2)2^{p+1}}}\right] \\
&\leq \mathbb{P}\left[\left|\frac{1}{M-1}\sum_{j=1}^{M-1}\widetilde{\mu}_k^{j}(p)-\mu_k\right| \geq 2\sqrt{\frac{\log(T)}{(M-1)2^{p+1}}}\right] \\
&\qquad + \mathbb{P}\left[\left|\widetilde{\mu}_k^{j}(p)-\mu_k\right| \geq 2\sqrt{\frac{\log(T)}{2^{p+1}}}\right] \\
&\leq \frac{6}{T}. \qquad (10)
\end{aligned}
$$

Moreover by construction, no matter what mean sent the selfish player,

$$
\min_{j\in[M-1]}\frac{1}{M-2}\sum_{\substack{j'\in[M-1]\\ j'\neq j}}\widetilde{\mu}_k^{j}(p) \leq \widetilde{\mu}_k(p) \leq \max_{j\in[M-1]}\frac{1}{M-2}\sum_{\substack{j'\in[M-1]\\ j'\neq j}}\widetilde{\mu}_k^{j}(p).
$$

Indeed, assume that the selfish player sends a mean larger than any other player. Then her mean as well as the minimal sent mean are cut out and $\widetilde{\mu}_k(p)$ is then equal to the right term. Conversely if she sends the smallest mean, $\widetilde{\mu}_k(p)$ corresponds to the left term. Since $\widetilde{\mu}_k(p)$ is non-decreasing in $\widetilde{\mu}_k^M(p)$, the inequality also holds in the case where the selfish player sends neither the smallest nor the largest mean.

Finally, using a union bound over all $j \in [M-1]$ with equation (10) yields Lemma 17. ■

Using classical MAB techniques then yields Proposition 1.

**Proposition 1.** *Independently of the selfish player behavior, as long as the* `PunishHomogeneous` *protocol is not used, with probability* $1-\mathcal{O}\left(\frac{KM\log(T)}{T}\right)$*, every optimal arm $k$ is accepted after at most* $\mathcal{O}\left(\frac{\log(T)}{(\mu_k-\mu_{(M+1)})^2}\right)$ *pulls and every sub-optimal arm $k$ is rejected after at most* $\mathcal{O}\left(\frac{\log(T)}{(\mu_{(M)}-\mu_k)^2}\right)$ *pulls during exploration phases.*

**Proof.** The fact that the `PunishHomogeneous` protocol is not started just means that no corrupted message is sent between cooperative players. The proof is conditioned on the success of the initialization phase, which happens with probability $1-\mathcal{O}\left(\frac{M}{T}\right)$. Note that there are at most $\log_2(T)$ exploration phases. Thanks to Lemma 17, with probability $1-\mathcal{O}\left(\frac{KM\log(T)}{T}\right)$, the inequality $|\widetilde{\mu}_k(p)-\mu_k| \leq B(p)$ thus holds for any $p$. The remaining of the proof is conditioned on this event. Especially, an optimal arm is never rejected and a suboptimal one never accepted.

First consider an optimal arm $k$ and note $\Delta_k = \mu_k - \mu_{(M+1)}$ the optimality gap. Let $p_k$ be the smallest integer $p$ such that $(M-2)2^{p+1} \geq \frac{16^2 \log(T)}{\Delta_k^2}$. In particular, $4B(p_k) \leq \Delta_k$, which implies that the arm $k$ is accepted at the end of the communication phase $p_k$ or before.

Necessarily, $(M-2)2^{p_k+1} \leq \frac{2 \cdot 16^2 \log(T)}{\Delta_k^2}$ and especially, $M2^{p_k+1} = \mathcal{O}\left(\frac{\log(T)}{\Delta_k^2}\right)$. Note that the number of exploratory pulls on arm $k$ during the $p$ first phases is bounded by $M(2^{p+1}+p)$[8], leading to Proposition 1. The same holds for the sub-optimal arms with $\Delta_k = \mu_{(M)} - \mu_k$. ■

In the following, we keep the notation $t_k = \frac{c\log(T)}{\left(\mu_k - \mu_{(M)}\right)^2}$, where $c$ is a universal constant, such that with probability $1 - \mathcal{O}\left(\frac{KM}{T}\right)$, any arm $k$ is correctly accepted or rejected after a time at most $t_k$. All players are now assumed to play `SIC-GT`, e.g., there is no selfish player. Since there is no collision during exploration/exploitation (conditionally on the success of the initialization phase), the following decomposition holds (Anantharam et al., 1987):

$$R^{\text{explo}} = \sum_{k>M} (\mu_{(M)} - \mu_{(k)}) T_{(k)}^{\text{explo}} + \sum_{k \leq M} (\mu_{(k)} - \mu_{(M)})(T^{\text{explo}} - T_{(k)}^{\text{explo}}), \tag{11}$$

where $T^{\text{explo}} = \#\text{Explo}$ and $T_{(k)}^{\text{explo}}$ is the centralized number of pulls on the $k$-th best arm during exploration or exploitation.

**Lemma 18.** *If all players follow* `SIC-GT`*, with probability* $1 - \mathcal{O}\left(\frac{KM\log(T)}{T}\right)$*, it holds:*

- *for* $k > M$, $(\mu_{(M)} - \mu_{(k)}) T_{(k)}^{\textit{explo}} = \mathcal{O}\left(\frac{\log(T)}{\mu_{(M)} - \mu_{(k)}}\right)$.
- $\sum_{k \leq M} (\mu_{(k)} - \mu_{(M)})(T^{\textit{explo}} - T_{(k)}^{\textit{explo}}) = \mathcal{O}\left(\sum_{k>M} \frac{\log(T)}{\mu_{(M)} - \mu_k}\right)$.

**Proof.** With probability $1 - \mathcal{O}\left(\frac{KM\log(T)}{T}\right)$, Proposition 1 yields that any arm $k$ is correctly accepted or rejected at time at most $t_k$. The remaining of the proof is conditioned on this event and the success of the initialization phase. The first point of Lemma 18 is a direct consequence of Proposition 1. It remains to prove the second point.

Let $\hat{p}_k$ be the number of the phase at which the arm $k$ is either accepted or rejected and let $K_p$ be the number of arms that still need to be explored at the beginning of phase $p$ and $M_p$ be the number of optimal arms that still need to be explored. The following two key Lemmas are crucial to obtain the second point.

**Lemma 19.** *Under the assumptions of Lemma 18:*

$$\sum_{k \leq M} (\mu_{(k)} - \mu_{(M)})(T^{\textit{explo}} - T_{(k)}^{\textit{explo}}) \leq \sum_{j>M} \sum_{k \leq M} \sum_{p=1}^{\min(\hat{p}_{(k)}, \hat{p}_{(j)})} (\mu_{(k)} - \mu_{(M)}) 2^p \frac{M}{M_p} + o(\log(T)).$$

**Lemma 20.** *Under the assumptions of Lemma 18, for any* $j > M$*:*

$$\sum_{k \leq M} \sum_{p=1}^{\min(\hat{p}_{(k)}, \hat{p}_{(j)})} (\mu_{(k)} - \mu_{(M)}) 2^p \frac{M}{M_p} \leq \mathcal{O}\left(\frac{\log(T)}{\mu_{(M)} - \mu_{(j)}}\right).$$

8. During the exploration phase $p$, any explored arm is pulled between $M2^p$ and $M(2^p+1)$ times.

Combining these two Lemmas with Equation (11) finally yields Lemma 15. ■

**Proof of Lemma 19.** Consider an optimal arm $k$. During the $p$-th exploration phase, either $k$ has already been accepted and is pulled $M\left\lceil \frac{K_p 2^p}{M_p}\right\rceil$ times; or $k$ has not been accepted yet and is pulled at least $2^p M$, i.e., is not pulled at most $M\left(\left\lceil \frac{K_p 2^p}{M_p}\right\rceil - 2^p\right)$ times. This gives:

$$
\begin{aligned}
(\mu_{(k)} - \mu_{(M)})(T^{\text{explo}} - T^{\text{explo}}_{(k)}) &\leq \sum_{p=1}^{\hat{p}_k} (\mu_{(k)} - \mu_{(M)}) M \left(\left\lceil \frac{K_p 2^p}{M_p}\right\rceil - 2^p\right), \\
&\leq \sum_{p=1}^{\hat{p}_k} (\mu_{(k)} - \mu_{(M)}) M \left(\frac{K_p 2^p}{M_p} - 2^p + 1\right), \\
&\leq \hat{p}_k (\mu_{(k)} - \mu_{(M)}) M + \sum_{p=1}^{\hat{p}_k} (\mu_{(k)} - \mu_{(M)})(K_p - M_p)\frac{M}{M_p} 2^p.
\end{aligned}
$$

We assumed that any arm $k$ is correctly accepted or rejected after a time at most $t_k$. This implies that $\hat{p}_k = o(\log(T))$. Moreover, $K_p - M_p$ is the number of suboptimal arms not rejected at phase $p$, i.e., $K_p - M_p = \sum_{j>M} \mathbb{1}_{p \leq \hat{p}_{(j)}}$ and this proves Lemma 19. ■

**Proof of Lemma 20.** For $j > M$, define $A_j = \sum_{k \leq M} \sum_{p=1}^{\min(\hat{p}_{(k)}, \hat{p}_{(j)})} (\mu_{(k)} - \mu_{(M)}) 2^p \frac{M}{M_p}$. We want to show $A_j \leq \mathcal{O}\left(\frac{\log(T)}{\mu_{(M)} - \mu_{(j)}}\right)$ with the considered conditions. Note $T(p) = M(2^{p+1} - 1)$ and $\Delta(p) = \sqrt{\frac{c \log(T)}{T(p)}}$. The inequality $\hat{p}_{(k)} \geq p$ then implies $\mu_{(k)} - \mu_{(M)} < \Delta(p)$, i.e.,

$$
\begin{aligned}
A_j &\leq \sum_{k \leq M} \sum_{p=1}^{\hat{p}_{(j)}} 2^p \Delta(p) \mathbb{1}_{p \leq \hat{p}_{(k)}} \frac{M}{M_p} = \sum_{p=1}^{\hat{p}_{(j)}} 2^p \Delta(p) M \\
&\leq \sum_{p=1}^{\hat{p}_{(j)}} \Delta(p)(T(p) - T(p-1))
\end{aligned}
$$

The equality comes because $\sum_{k \leq M} \mathbb{1}_{p \leq \hat{p}_{(k)}}$ is exactly $M_p$. Then from the definition of $\Delta(p)$:

$$
\begin{aligned}
A_j &\leq c \log(T) \sum_{p=1}^{\hat{p}_{(j)}} \Delta(p) \left(\frac{1}{\Delta(p)} + \frac{1}{\Delta(p-1)}\right)\left(\frac{1}{\Delta(p)} - \frac{1}{\Delta(p-1)}\right) \\
&\leq (1+\sqrt{2}) c \log(T) \sum_{p=1}^{\hat{p}_{(j)}} \left(\frac{1}{\Delta(p)} - \frac{1}{\Delta(p-1)}\right) \\
&\leq (1+\sqrt{2}) c \log(T) / \Delta(\hat{p}_{(j)}) \\
&\leq (1+\sqrt{2}) \sqrt{c \log(T) T(\hat{p}_{(j)})}
\end{aligned}
$$

By definition, $T(\hat{p}_{(j)})$ is smaller than the number of exploratory pulls on the $j$-th best arm and is thus bounded by $\frac{c\log(T)}{(\mu_{(M)}-\mu_{(j)})^2}$, leading to Lemma 20. ■

### C.3. Selfish robustness of SIC-GT

In this section, the second point of Theorem 6 is proven. First Lemma 21 gives guarantees for the punishment protocol. Its proof is given in Appendix C.3.1.

**Lemma 21.** *If the* `PunishHomogeneous` *protocol is started at time* $T_{\text{punish}}$ *by* $M-1$ *players, then for the remaining player* $j$*, independently of her sampling strategy:*

$$\mathbb{E}[\text{Rew}_T^j|\text{punish}] \leq \mathbb{E}[\text{Rew}^j_{T_{punish}+t_p}] + \widetilde{\alpha}\frac{T - T_{\text{punish}} - t_p}{M}\sum_{k=1}^{M}\mu_{(k)},$$

*with* $t_p = \mathcal{O}\left(\frac{K}{(1-\widetilde{\alpha})^2\mu_{(K)}}\log(T)\right)$ *and* $\widetilde{\alpha} = \frac{1+(1-1/K)^{M-1}}{2}$.

**Proof of the second point of Theorem 6 (Nash equilibrium).** First fix $T_{\text{punish}}$ the time at which the punishment protocol starts if it happens (and $T$ if it does not). Before this time, the selfish player can not perturb the initialization phase, except by changing the ranks distribution. Moreover, the exploration/exploitation phase is not perturbed as well, as claimed by Proposition 1. The optimal strategy then earns at most $T_{\text{init}}$ during the initialization and #Comm during the communication. With probability $1-\mathcal{O}\left(\frac{KM\log(T)}{T}\right)$, the initialization is successful and the concentration bound of Lemma 5 holds for any arm and player all the time. The following is conditioned on this event.

Note that during the exploration, the cooperative players pull any arm the exact same amount of times. Since the upper bound time $t_k$ to accept or reject an arm does not depend on the strategy of the selfish player, Lemma 18 actually holds for the cooperative player, i.e., for any cooperative player $j$:

$$\sum_{k\leq M}\left(\mu_{(k)}-\mu_{(M)}\right)\left(\frac{T^{\text{explo}}}{M} - T^j_{(k)}\right) = \mathcal{O}\left(\frac{1}{M}\sum_{k>M}\frac{\log(T)}{\mu_{(M)}-\mu_k}\right), \tag{12}$$

where $T^j_{(k)}$ is the number of pulls by player $j$ on the $k$-th best arm during the exploration/exploitation. The same kind of regret decomposition as in Equation (11) is possible for the regret of the selfish player $j$ and especially:

$$R_j^{\text{explo}} \geq \sum_{k\leq M}(\mu_{(k)}-\mu_{(M)})\left(\frac{T^{\text{explo}}}{M} - T^j_{(k)}\right).$$

However, the optimal strategy for the selfish player is to pull the best available arm during the exploration and especially to avoid collisions. This implies the constraint $T^j_{(k)} \leq T^{\text{explo}} - \sum_{j\neq j'} T^{j'}_{(k)}$. Using this constraint with Equation (12) yields $\frac{T^{\text{explo}}}{M} - T^j_{(k)} \geq -\sum_{j\neq j'}\frac{T^{\text{explo}}}{M} - T^{j'}_{(k)}$ and then

$$R_j^{\text{explo}} \geq -\mathcal{O}\left(\sum_{k>M}\frac{\log(T)}{\mu_{(M)}-\mu_k}\right),$$

which can be rewritten as

$$\mathrm{Rew}_j^{\mathrm{explo}} \leq \frac{T^{\mathrm{explo}}}{M} \sum_{k=1}^{M} \mu_{(k)} + \mathcal{O}\left( \sum_{k>M} \frac{\log(T)}{\mu_{(M)} - \mu_k} \right).$$

Thus, for any strategy $s'$ when adding the low probability event of a failed exploration or initialization,

$$\begin{aligned}\mathbb{E}[\mathrm{Rew}^j_{t_p+T_{\mathrm{punish}}}(s', s_{-j})] \leq {} & (T_{\mathrm{init}} + \#\mathrm{Comm} + t_p + \mathcal{O}(KM\log(T))) \\ & + \frac{\mathbb{E}[T_{\mathrm{punish}}] - T_{\mathrm{init}} - \#\mathrm{Comm}}{M} \sum_{k\leq M} \mu_{(k)} + \mathcal{O}\left( \sum_{k>M} \frac{\log(T)}{\mu_{(M)} - \mu_k} \right).\end{aligned}$$

Using Lemma 21, this yields:

$$\begin{aligned}\mathbb{E}[\mathrm{Rew}^j_T(s', s_{-j})] \leq {} & (T_{\mathrm{init}} + \#\mathrm{Comm} + t_p + \mathcal{O}(KM\log(T))) \\ & + \frac{\mathbb{E}[T_{\mathrm{punish}}] - T_{\mathrm{init}} - \#\mathrm{Comm}}{M} \sum_{k\leq M} \mu_{(k)} + \mathcal{O}\left( \sum_{k>M} \frac{\log(T)}{\mu_{(M)} - \mu_k} \right) \\ & + \widetilde{\alpha} \frac{T - \mathbb{E}[T_{\mathrm{punish}}]}{M} \sum_{k=1}^{M} \mu_{(k)}.\end{aligned}$$

The right term is maximized when $\mathbb{E}[T_{\mathrm{punish}}]$ is maximized, i.e., when it is $T$. We then get:

$$\mathbb{E}[\mathrm{Rew}^j_T(s', s_{-j})] \leq \frac{T}{M} \sum_{k\leq M} \mu_{(k)} + \varepsilon,$$

where $\varepsilon = \mathcal{O}\left( \sum_{k>M} \frac{\log(T)}{\mu_{(M)}-\mu_k} + K^2 \log(T) + MK \log^2\left( \frac{\log(T)}{(\mu_{(M)}-\mu_{(M+1)})^2} \right) + \frac{K\log(T)}{(1-\widetilde{\alpha})^2 \mu_{(K)}} \right)$. ■

**Proof of the second point of Theorem 6 (stability).** Define $\mathcal{E}$ the *bad event* that the initialization is not successful or that an arm is poorly estimated at some time. Let $\varepsilon' = T\mathbb{P}[\mathcal{E}] + \mathbb{E}[\#\mathrm{Comm}|\neg\mathcal{E}] + K\log(T)$. Then $\varepsilon' = \mathcal{O}\left( KM\log(T) + KM\log^2\left( \frac{\log(T)}{(\mu_{(M)}-\mu_{(M+1)})^2} \right) \right)$.

Assume that the player $j$ is playing a deviation strategy $s'$ such that for some other player $i$ and $l > 0$:

$$\mathbb{E}[\mathrm{Rew}^i_T(s', s_{-j})] \leq \mathbb{E}[\mathrm{Rew}^i_T(s)] - l - \varepsilon'$$

First fix $T_{\mathrm{punish}}$ the time at which the punishment protocol starts. Let us now compare $s'$ with the individual optimal strategy for player $j$, $s^*$. Let $\varepsilon'$ take account of the communication phases, the initialization and the low probability events.

The number of pulls by any player during exploration/exploitation is given by Equation (12) unless the punishment protocol is started. Moreover, the selfish player causes at most a collision during exploration/exploitation before initiating the punishment protocol, so the loss of player $i$ before punishment is at most $1 + \varepsilon'$.

After $T_{\mathrm{punish}}$, Lemma 21 yields that the selfish player suffers a loss at least $(1-\widetilde{\alpha})\frac{T-T_{\mathrm{punish}}-t_p}{M} \sum_{k=1}^{M} \mu_{(k)}$, while any cooperative player suffers at most $\frac{T-T_{\mathrm{punish}}}{M} \sum_{k=1}^{M} \mu_{(k)}$.

The selfish player then suffers after $T_{\text{punish}}$ a loss at least $(1-\widetilde{\alpha})((l-1)-t_p)$. Define $\beta = 1-\widetilde{\alpha}$. We just showed:

$$\mathbb{E}[\text{Rew}_T^i(s', s_{-j})] \leq \mathbb{E}[\text{Rew}_T^i(s)] - l - \varepsilon' \implies \mathbb{E}[\text{Rew}_T^j(s', s_{-j})] \leq \mathbb{E}[\text{Rew}_T^j(s^*, s_{-j})] - \beta(l-1) + \beta t_p$$

Moreover, thanks to the second part of Theorem 6, $\mathbb{E}[\text{Rew}_T^j(s^*, s_{-j})] \leq \mathbb{E}[\text{Rew}_T^j(s)] + \varepsilon$ with $\varepsilon = \mathcal{O}\left(\sum_{k>M} \frac{\log(T)}{\mu_{(M)}-\mu_k} + K^2 \log(T) + MK \log^2\left(\frac{\log(T)}{(\mu_{(M)}-\mu_{(M+1)})^2}\right) + \frac{K\log(T)}{(1-\widetilde{\alpha})^2 \mu_{(K)}}\right)$. Then by defining $l_1 = l + \varepsilon'$, $\varepsilon_1 = \varepsilon + \beta t_p + \beta\varepsilon' + 1 = \mathcal{O}(\varepsilon)$, we get:

$$\mathbb{E}[\text{Rew}_T^i(s', s_{-j})] \leq \mathbb{E}[\text{Rew}_T^i(s)] - l_1 \implies \mathbb{E}[\text{Rew}_T^j(s', s_{-j})] \leq \mathbb{E}[\text{Rew}_T^j(s)] + \varepsilon_1 - \beta l_1.$$

■

### C.3.1. PROOF OF LEMMA 21.

The punishment protocol starts by estimating all means $\mu_k$ with a multiplicative precision of $\delta$. This is possible thanks to Lemma 22, which corresponds to Theorem 9 in (Cesa-Bianchi et al., 2019) and Lemma 13 in (Berthet and Perchet, 2017).

**Lemma 22.** *Let $X_1, \ldots, X_n$ be $n$-i.i.d. random variables in $[0,1]$ with expectation $\mu$ and define $S_t^2 = \frac{1}{t-1}\sum_{s=1}^t (X_s - \bar{X}_t)^2$. For all $\delta \in (0,1)$, if $n \geq n_0$, where*

$$n_0 = \left\lceil \frac{2}{3\delta\mu} \log(T) \left(\sqrt{9\frac{1}{\delta^2} + 96\frac{1}{\delta} + 85} + \frac{3}{\delta} + 1\right)\right\rceil + 2 = \mathcal{O}\left(\frac{1}{\delta^2\mu}\log(T)\right)$$

*and $\tau$ is the smallest time $t \in \{2, \ldots, n\}$ such that*

$$\delta\bar{X}_t \geq 2S_t\left(\log(T)/t\right)^{1/2} + \frac{14\log(T)}{3(t-1)},$$

*then, with probability at least $1 - \frac{3}{T}$:*

1. *$\tau \leq n_0$,*
2. *$(1-\delta)\,\bar{X}_\tau < \mu < (1+\delta)\,\bar{X}_\tau$.*

**Proof of Lemma 21**

The punishment protocol starts for all cooperative players at $T_{\text{punish}}$. For $\delta = \frac{1-\gamma}{1+3\gamma}$, each player then estimates each arm. Lemma 22 gives that with probability at least $1-3/T$:

- the estimation ends after a time at most $t_p = \mathcal{O}\left(\frac{K}{\delta^2\mu_{(K)}}\log(T)\right)$,
- $(1-\delta)\widehat{\mu}_k^j \leq \mu_k \leq (1+\delta)\widehat{\mu}_k^j$.

The following is conditioned on this event. The last inequality can be reversed as $\frac{\mu_k}{1+\delta} \leq \widehat{\mu}_k^j \leq \frac{\mu_k}{1-\delta}$. Then, this implies for any cooperative player $j$

$$1 - p_k^j \leq \left( \gamma \frac{(1+\delta) \sum_{m=1}^M \mu_{(m)}}{(1-\delta) M \mu_k} \right)^{\frac{1}{M-1}}.$$

The expected reward that gets the selfish player $j$ by pulling $k$ after the time $T_{\text{punish}} + t_p$ is thus smaller than $\gamma \frac{1+\delta}{1-\delta} \frac{\sum_{m=1}^M \mu_{(m)}}{M}$.

Note that $\gamma \frac{1+\delta}{1-\delta} = \frac{1+\gamma}{2} = \widetilde{\alpha}$. Considering the low probability event given by Lemma 22 adds a constant term that can be counted in $t_p$. This finally yields the result of Lemma 21. ■

## Appendix D. Supplementary material for RSD-GT

### D.1. Description of the algorithm

This section provides a complete description of RSD-GT. Its pseudocode is given in Algorithm 4. It relies on auxiliary protocols described by Protocols 3, 10, 11, 12, 13 and 14.

**Initialization phase.** RSD-GT starts with the exact same initialization as SIC-GT, which is given by Protocol 3, to estimate $M$ and attribute ranks among the players. Afterwards, they start the exploration.

In the remaining of the algorithm, as already explained in Section 5.3, the time is divided into superblocks, which are divided into $M$ blocks of length $5K + MK + M^2K$. During the $j$-th block of a superblock, the dictators ordering for RSD is $(j, \dots, M, 1, \dots, j-1)$. Moreover, only the $j$-th player can send messages during this block if she is still exploring.

**Exploration.** The exploiting players sequentially pull all the arms in $[K]$ to avoid collisions with any other exploring player. Yet, they still collide with exploiting players.

RSD-GT is designed so that all players know at each round the $M$ preferred arms of any exploiting players and their order. The players thus know which arms are occupied by the exploiting players during a block $j$. The communication arm is thus a common arm unoccupied by any exploiting player. When an exploring player encounters a collision on this arm at the beginning of the block, this means that another player signaled the start of a communication block. In that case, the exploring player starts `Listen`, described by Protocol 11, to receive the messages of the communicating player.

On the other hand, when an exploring player $j$ knows her $M$ preferred arms and their order, she waits for the next block $j$ to initiate communication. She then proceeds to `SignalPreferences`, given by Protocol 13.

**Communication block.** In a communication block, the communicating player first collides with each exploiting and exploring player to signal them the start of a communication block as described by Protocol 12. These collisions need to be done in a particular way given by `SendBit` so that all players correctly detect the start of a communication block. These players then repeat this signal to ensure that every player is listening.

**Algorithm 4:** `RSD-GT`

**Input:** $T, \delta$
$\widehat{M}, j \leftarrow$ `Initialize` $(T, K)$; state $\leftarrow$ "exploring" and blocknumber $\leftarrow 1$
Let $\boldsymbol{\pi}$ be a $M \times M$ matrix with only 0 // $\pi_k^j$ is the $k$-th preferred arm by $j$
**while** $t < T$ **do**
  blocktime $\leftarrow t \pmod{5K + MK + M^2K} + 1$
  **if** *blocktime* $= 1$ **then** // new block
    blocknumber $\leftarrow$ blocknumber $\pmod{M} + 1$; $b_k^j(t) \leftarrow \sqrt{2\log(T)/T_k^j(t)}$
    Let $\lambda^j$ be the ordering of the empirical means: $\widehat{\mu}^j_{\lambda^j_k}(t) \geq \widehat{\mu}^j_{\lambda^j_{k+1}}(t)$ for any $k$
    **if** *(blocknumber, state)* $= (j,$ *"exploring"*$)$ *and* $\forall k \in [M], \hat{\mu}^j_{\lambda^j_k} - b^j_{\lambda^j_k} \geq \hat{\mu}^j_{\lambda^j_{k+1}} + b^j_{\lambda^j_{k+1}}$
     **then** $\pi^j \leftarrow \lambda^j$; state $\leftarrow$ `SignalPreferences` $(\boldsymbol{\pi}, j)$ // send Top-M arms
  **end**
  $(l,$ comm_arm$) \leftarrow$ `ComputeRSD` $(\boldsymbol{\pi},$ blocknumber$)$ // $j$ pulls $l^j$
  **if** *state* $=$ *"exploring"* **then**
    Pull $l^j$ and update $\widehat{\mu}^j_{l^j}$
    **if** $l^j =$ *comm_arm and* $\eta_{l^j} = 1$ **then** // received signal
      **if** *blocktime* $> 4K$ **then** state $\leftarrow$ "punishing"
      **else** (state, $\pi^{\text{blocknumber}}$) $\leftarrow$ `Listen` (blocknumber, state, $\boldsymbol{\pi}$, comm_arm)
  **end**
  **if** *state* $=$ *"exploiting" and* $\exists i, k$ *such that* $\pi_k^i = 0$ **then**
    Pull $l^j$ // arm attributed by RSD algo
    **if** $l^j \notin \{l^i | i \in [M] \setminus \{j\}\}$ *and* $\eta_{l^j}(t) = 1$ **then** // received signal
      **if** *blocktime* $> 4K$ **then** state $\leftarrow$ "punishing"
      **else** (state, $\pi^{\text{blocknumber}}$) $\leftarrow$ `Listen` (blocknumber, state, $\boldsymbol{\pi}$, comm_arm)
  **end**
  **if** *state* $=$ *"exploiting" and* $\forall i, k, \pi_k^i \neq 0$ **then** // all players are exploiting
    Draw inspect $\sim$ Bernoulli$(\sqrt{\log(T)}/T)$
    **if** *inspect* $= 1$ **then** // random inspection
      Pull $l^i$ with $i$ chosen uniformly at random among the other players
      **if** $\eta_{l^i} = 0$ **then** state $\leftarrow$ "punishing" // lying player
    **else**
      Pull $l^j$; **if** *observed two collisions in a row* **then** state $\leftarrow$ "punishing"
  **end**
  **if** *state* $=$ *"punishing"* **then** `PunishSemiHetero` $(\delta)$
**end**

The communicating player then sends to all players her $M$ preferred arms in order of preferences. Afterwards, each player repeats this list to ensure that no malicious player interfered during communication. As soon as some malicious behavior is observed, the start of `PunishSemiHetero`, given by Protocol 14, is signaled to all players.

**Protocol 10:** `ComputeRSD`

```
Input: π, blocknumber
taken_arms ← ∅
for s = 0, ..., M − 1 do
    dict ← s + blocknumber − 1(mod M) + 1                    // current dictator
    p ← min{p' ∈ [M] | π_{p'}^dict ∉ taken_arms}             // best available choice
    if π_p^dict ≠ 0 then l^dict ← π_p^dict and add π_p^dict to taken_arms
    else l^dict ← t + dict (mod K) + 1                        // explore
end
comm_arm ← min[K] \ taken_arms
return (l, comm_arm)
```

**Exploitation.** An exploiting player starts any block $j$ by computing the attribution of the RSD algorithm between the exploiting players given their known preferences and the dictatorship ordering $(j, \dots, j-1)$. She then pulls her attributed arm for the whole block, unless she receives a signal.

A signal is received when she collides with an exploring player, while unintended[9]. If it is at the beginning of a block, it means that a communication block starts. Otherwise, she just enters the punishment protocol. Note that the punishment protocol starts by signaling the start of `PunishSemiHetero` to ensure that every cooperative player starts punishing.

Another security is required to ensure that the selfish player truthfully reports her preferences. She could otherwise report fake preferences to decrease another player's utility while her best arm remains uncontested and thus available. To avoid this, `RSD-GT` uses *random inspections* when all players are exploiting. With probability $\sqrt{\log(T)}/T$ at each round, any player checks that some other player is indeed exploiting the arm she is attributed by the RSD algorithm. If it is not the case, the inspecting player signals the start of `PunishSemiHetero` to everyone by colliding twice with everybody, since a single collision could be a random inspection. Because of this, the selfish player can not pull another arm than the attributed one too often without starting a punishment scheme. Thus, if she did not report her preferences truthfully, this also has a cost for her.

9. She normally collides with exploring players. Yet as she knows the set of exploring players, she exactly knows when this happens.

**Protocol 11:** `Listen`

**Input:** blocknumber, state, $\boldsymbol{\pi}$, arm_comm
ExploitPlayers $= \{i \in [M] \mid \pi_1^i \neq 0\}$; $\lambda \leftarrow \pi^{\text{blocknumber}}$
**if** $\lambda_1 \neq 0$ **then** state $\leftarrow$ "punishing" `// this player already sent`
**while** *blocktime* $\leq 2K$ **do** Pull $t + j (\text{mod } K) + 1$
**if** *blocktime* $= 2K$ **then** `SendBit` (comm_arm, ExploitPlayers, $j$) `// repeat signal`
**else while** *blocktime* $\leq 4K$ **do** Pull $t + j (\text{mod } K) + 1$

**for** $K$ *rounds* **do**
  **if** *state = "punishing"* **then** Pull $j$ `// signal punishment`
  **else**
    Pull $k = t + j (\text{mod } K) + 1$ ; **if** $\eta_k = 1$ **then** state $\leftarrow$ "punishing"
**end**

**for** $n = 1, \dots, MK$ **do** `// receive preferences`
  Pull $k = t + j (\text{mod } K) + 1$
  $m \leftarrow \lceil n/K \rceil$ `// communicating player sends her m-th pref. arm`
  **if** $\eta_k = 1$ **then**
    **if** $\lambda_m \neq 0$ **then** state $\leftarrow$ "punishing" `// received two signals`
    **else** $\lambda_m \leftarrow k$
**end**

**for** $n = 1, \dots, M^2 K$ **do** `// repetition block`
  $m \leftarrow \left\lceil \frac{n \,(\text{mod } MK)}{K} \right\rceil$ and $l \leftarrow \lceil \frac{n}{MK} \rceil$ `// l repeats the m-th pref.`
  **if** $j = l$ **then** Pull $\lambda_m$
  **else**
    Pull $k = t + j \ (\text{mod } K) + 1$
    **if** $\eta_k = 1$ *and* $\lambda_m \neq k$ **then** state $\leftarrow$ "punishing" `// info differs`
**end**
**if** $\#\{\lambda_m \neq 0 \mid m \in [M]\} \neq M$ **then** state $\leftarrow$ "punishing" `// did not send all`
**return** (state, $\lambda$)

**Protocol 12:** `SendBit`

**Input:** comm_arm, ExploitPlayers, $j$
**if** *ExploitPlayers* $= \emptyset$ **then** $\widetilde{j} \leftarrow j$
**else** $\widetilde{j} \leftarrow \min$ ExploitPlayers
**for** $K$ *rounds* **do** Pull $t + \widetilde{j} (\text{mod } K) + 1$ `// send bit to exploiting players`
**for** $K$ *rounds* **do** Pull comm_arm `// send bit to exploring players`

### D.2. Regret analysis

This section aims at proving the first point of Theorem 7. `RSD-GT` uses the exact same initialization phase as `SIC-GT`, and its guarantees are thus given by Lemma 14. Here again, the regret is

**Protocol 13:** `SignalPreferences`

**Input:** $\boldsymbol{\pi}, j,$ comm_arm
ExploitPlayers $= \{i \in [M] \setminus \{j\} \mid \pi_1^i \neq 0\}$; $\lambda \leftarrow \pi^j$ // $\lambda$ is signal to send
state ← "exploiting" // state after the protocol
`SendBit` (comm_arm, ExploitPlayers, $j$) // initiate communication block
**for** $2K$ *rounds* **do** Pull $t + j \pmod K + 1$ // wait for repetition
**for** $K$ *rounds* **do** // receive punish signal
  Pull $t + j \pmod K + 1$; **if** $\eta_k = 1$ **then** state ← "punishing"
**end**
**for** $n = 1, \ldots, MK$ **do** pull $\lambda_{\lceil \frac{n}{K} \rceil}$ // send $k$-th preferred arm
**for** $n = 1, \ldots, M^2K$ **do** // repetition block
  $m \leftarrow \left\lceil \frac{n \pmod{MK}}{K} \right\rceil$ and $l \leftarrow \lceil \frac{n}{MK} \rceil$ // $l$ repeats the $m$-th pref.
  **if** $j = l$ **then** Pull $\lambda_m$
  **else**
    Pull $k = t + j \pmod K + 1$
    **if** $\eta_k = 1$ *and* $\lambda_m \neq k$ **then** state ← "punishing" // info differs
**end**
**return** state

decomposed into three parts:

$$R_T^{\text{RSD}} = R^{\text{init}} + R^{\text{comm}} + R^{\text{explo}}, \tag{13}$$

where
$$\begin{cases} R^{\text{init}} = T_{\text{init}} \mathbb{E}_{\sigma \sim \mathcal{U}(\mathfrak{S}_M)} \left[ \sum_{k=1}^{M} \mu_{\pi_\sigma(k)}^{\sigma(k)} \right] - \mathbb{E}_\mu \left[ \sum_{t=1}^{T_{\text{init}}} \sum_{j=1}^{M} r^j(t) \right] \text{ with } T_{\text{init}} = (12eK^2 + K)\log(T), \\ R^{\text{comm}} = \#\text{Comm} \mathbb{E}_{\sigma \sim \mathcal{U}(\mathfrak{S}_M)} \left[ \sum_{k=1}^{M} \mu_{\pi_\sigma(k)}^{\sigma(k)} \right] - \mathbb{E}_\mu \left[ \sum_{t \in \text{Comm}} \sum_{j=1}^{M} r^j(t)) \right], \\ R^{\text{explo}} = \#\text{Explo} \mathbb{E}_{\sigma \sim \mathcal{U}(\mathfrak{S}_M)} \left[ \sum_{k=1}^{M} \mu_{\pi_\sigma(k)}^{\sigma(k)} \right] - \mathbb{E}_\mu \left[ \sum_{t \in \text{Explo}} \sum_{j=1}^{M} r^j(t)) \right] \end{cases}$$

with Comm defined as all the rounds of a block where at least a cooperative player uses `Listen` protocol and Explo $= \{T_{\text{init}} + 1, \ldots, T\} \setminus$ Comm. In case of a successful initialization, a single player can only initiate a communication block once without starting a punishment protocol. Thus, as long as no punishment protocol is started: $\#\text{Comm} \leq M(5K + MK + M^2K) = \mathcal{O}(M^3K)$.

Denote by $\Delta^j = \min_{k \in [M]} \mu_{(k)}^j - \mu_{(k+1)}^j$ the level of precision required for player $j$ to know her $M$ preferred arms and their order. Proposition 2 gives the exploration time required for any player $j$:

**Proposition 2.** *With probability $1 - \mathcal{O}\left(\frac{K}{T}\right)$ and as long as no punishment protocol is started, the player $j$ starts exploiting after at most $\mathcal{O}\left(\frac{K\log(T)}{(\Delta^j)^2} + M^3K\right)$ exploration pulls.*

**Protocol 14:** `PunishSemiHetero`

**Input:** $\delta$

**if** *ExploitPlayers* $= [M]$ **then** collide with each player twice

**else** `// signal punishment during rounds` $3K+1, \dots, 5K$ `of a block`

  **for** $3K$ *rounds* **do** Pull $t + j (\mathrm{mod}\ K) + 1$

  `SendBit` (comm_arm, ExploitPlayers, $j$)

**end**

$\alpha \leftarrow \left(\frac{1+\delta}{1-\delta}\right)^2 (1 - 1/K)^{M-1}$ and $\delta' = \frac{1-\alpha}{1+3\alpha}$

Set $\widehat{\mu}_k^j, S_k^j, v_k^j, n_k^j \leftarrow 0$

**while** $\exists k \in [K], \delta' \widehat{\mu}_k^j < 2 s_k^j (\log(T)/n_k^j)^{1/2} + \frac{14 \log(T)}{3(n_k^j - 1)}$ **do** `// estimate` $\mu_k^j$

  Pull $k = t + j\ (\mathrm{mod}\ K) + 1$

  **if** $\delta' \widehat{\mu}_k^j < 2 s_k^j (\log(T)/n_k^j)^{1/2} + \frac{14 \log(T)}{3(n_k^j - 1)}$ **then**

    Update $\widehat{\mu}_k^j \leftarrow \frac{n_k^j}{n_k^j + 1} \widehat{\mu}_k^j + X_k(t)$ and $n_k^j \leftarrow n_k^j + 1$

    Update $S_k^j \leftarrow S_k^j + X_k^2$ and $s_k^j \leftarrow \sqrt{\frac{S_k^j - (\widehat{\mu}_k^j)^2}{n_k^j - 1}}$

**end**

$p_k \leftarrow \left(1 - \left(\alpha \frac{\sum_{l=1}^M \widehat{\mu}_{(l)}^j(t)}{M \widehat{\mu}_k^j(t)}\right)^{\frac{1}{M-1}}\right)_+$ ; $\widetilde{p}_k \leftarrow p_k / \sum_{l=1}^K p_l$ `// renormalize`

**while** $t \leq T$ **do** Pull $k$ with probability $p_k$ `// punish`

**Proof.** In the following, the initialization is assumed to be successful, which happens with probability $1 - \mathcal{O}\left(\frac{M}{T}\right)$. Moreover, Hoeffding inequality yields:

$$\mathbb{P}\left[\forall t \leq T, \left|\widehat{\mu}_k^j(t) - \mu_k^j(t)\right| \geq \sqrt{\frac{2\log(T)}{T_k^j(t)}}\right] \leq \frac{2}{T}$$

where $T_k^j(t)$ is the number of exploratory pulls on arm $k$ by player $j$. With probability $1 - \mathcal{O}\left(\frac{K}{T}\right)$, player $j$ then correctly estimates all arms at each round. The remaining of the proof is conditioned on this event.

During the exploration, player $j$ sequentially pulls the arms in $[K]$. Denote by $n$ the smallest integer such that $\sqrt{\frac{2\log(T)}{n}} \leq 4\Delta^j$. It directly comes that $n = \mathcal{O}\left(\frac{\log(T)}{(\Delta^j)^2}\right)$. Under the considered events, player $j$ then has determined her $M$ preferred arms and their order after $Kn$ exploratory pulls. Moreover, she needs at most $M$ blocks before being able to initiate her communication block and starts exploiting. Thus, she needs at most $\mathcal{O}\left(\frac{K\log(T)}{(\Delta^j)^2} + M^3 K\right)$ exploratory pulls, leading to Proposition 2. ■

**Proof of the first point of Theorem 7.** Assume all players play `RSD-GT`. Simply by bounding the size of the initialization and the communication phases, it comes:

$$R^{\text{init}} + R^{\text{comm}} \leq \mathcal{O}\left(MK^2 \log(T)\right).$$

Proposition 2 yields that with probability $1 - \mathcal{O}\left(\frac{KM}{T}\right)$, all players start exploitation after at most $\mathcal{O}\left(\frac{K\log(T)}{\Delta^2}\right)$ exploratory pulls.

For $p = \sqrt{\log(T)}/T$, with probability $\mathcal{O}(p^2 M)$ at any round $t$, a player is inspecting another player who is also inspecting or a player receives two consecutive inspections. These are the only ways to start punishing when all players are cooperative. As a consequence, when all players follow RSD-GT, they initiate the punishment protocol with probability $\mathcal{O}\left(p^2 MT\right)$. Finally, the total regret due to this event grows as $\mathcal{O}\left(M^2\log(T)\right)$.

If the punishment protocol is not initiated, players cycle through the RSD matchings of $\sigma \circ \sigma_0^{-1}, \ldots, \sigma \circ \sigma_0^{-M}$ where $\sigma_0$ is the classical $M$-cycle and $\sigma$ is the players permutation returned by the initialization. Define $U(\sigma) = \sum_{k=1}^M \mu^{\sigma(k)}_{\pi_\sigma(k)}$, where $\pi_\sigma(k)$ is the arm attributed to the $k$-th dictator, $\sigma(k)$, as defined in Section 4.2.2. $U(\sigma)$ is the social welfare of RSD algorithm when the dictatorships order is given by the permutation $\sigma$. As players all follow RSD-GT here, $\sigma$ is chosen uniformly at random in $\mathfrak{S}_M$ and any $\sigma \circ \sigma_0^{-k}$ as well. Then

$$\mathbb{E}_{\sigma\sim\mathcal{U}(\mathfrak{S}_M)}\left[\frac{1}{M}\sum_{k=1}^{M} U(\sigma\circ\sigma_0^{-M})\right] = \mathbb{E}_{\sigma\sim\mathcal{U}(\mathfrak{S}_M)}\left[U(\sigma)\right].$$

This means that in expectation, the utility given by the exploitation phase is the same as the utility of the RSD algorithm when choosing a permutation uniformly at random. Considering the low probability event of a punishment protocol, an unsuccesful initialization or a bad estimation of an arm finally yields:

$$R^{\text{explo}} \leq \mathcal{O}\left(\frac{MK\log(T)}{\Delta^2}\right).$$

Equation (13) concludes the proof. ■

### D.3. Selfish-robustness of RSD-GT

In this section, we prove the two last points of Theorem 7. Three auxiliary Lemmas are first needed. They are proved in Appendix D.3.1.

1. Lemma 23 compares the utility received by player $j$ from the RSD algorithm with the utility given by sequentially pulling her $M$ best arms in the $\delta$-heterogeneous setting.

2. Lemma 24 gives an equivalent version of Lemma 21, but for the $\delta$-heterogeneous setting.

3. Lemma 25 states that the expected utility of the assignment of any player during the exploitation phase does not depend on the strategy of the selfish player. The intuition behind this result is already given in Section 5.3.

   In the case of several selfish players, they could actually fix the joint distribution of $(\sigma^{-1}(j), \sigma^{-1}(j'))$. A simple rotation with a $M$-cycle is then not enough to recover a uniform distribution over $\mathfrak{S}_M$ in average. A more complex rotation is then required and the dependence in $M$ would blow up with the number of selfish players.

**Lemma 23.** *In the $\delta$-heterogeneous case for any player $j$ and permutation $\sigma$:*

$$\frac{1}{M}\sum_{k=1}^{M}\mu^j_{(k)} \leq \widetilde{U}_j(\sigma) \leq \frac{(1+\delta)^2}{(1-\delta)^2 M}\sum_{k=1}^{M}\mu^j_{(k)},$$

*where* $\widetilde{U}_j(\sigma) \coloneqq \frac{1}{M}\sum_{k=1}^{M}\mu^j_{\pi_{\sigma\circ\sigma_0^{-k}}\left(\sigma_0^k\circ\sigma^{-1}(j)\right)}$.

Following the notation of Section 4.2.2, $\pi_\sigma(\sigma^{-1}(j))$ is the arm attributed to player $j$ by RSD when the dictatorship order is given by $\sigma$. $\widetilde{U}_j(\sigma)$ is then the average utility of the exploitation when $\sigma$ is the permutation given by the initialization.

**Lemma 24.** *Recall that $\gamma = (1-1/K)^{M-1}$. In the $\delta$-heterogeneous setting with $\delta < \frac{1-\sqrt{\gamma}}{1+\sqrt{\gamma}}$, if the punish protocol is started at time $T_{punish}$ by $M-1$ players, then for the remaining player $j$, independently of her sampling strategy:*

$$\mathbb{E}[\mathrm{Rew}^j_T|\textit{punishment}] \leq \mathbb{E}[\mathrm{Rew}^j_{T_{punish}+t_p}] + \widetilde{\alpha}\frac{T-T_{punish}-t_p}{M}\sum_{k=1}^{M}\mu^j_{(k)},$$

*with* $t_p = \mathcal{O}\left(\frac{K\log(T)}{(1-\delta)(1-\widetilde{\alpha})^2\mu_{(K)}}\right)$ *and* $\widetilde{\alpha} = \frac{1+\left(\frac{1+\delta}{1-\delta}\right)^2\gamma}{2}$.

**Lemma 25.** *The initialization phase is successful when all players end with different ranks in $[M]$. For any player $j$, independently of the behavior of the selfish player:*

$$\mathbb{E}_{\sigma\sim\textit{successful initialization}}\left[\widetilde{U}_j(\sigma)\right] = \mathbb{E}_{\sigma\sim\mathcal{U}(\mathfrak{S}_M)}\left[\mu^j_{\pi_\sigma(\sigma^{-1}(j))}\right].$$

*where $\widetilde{U}_j(\sigma)$ is defined as in Lemma 23 above.*

**Proof of the second point of Theorem 7 (Nash equilibrium).** First fix $T_{\text{punish}}$ the beginning of the punishment protocol. Note $s$ the profile where all players follow `RSD-GT` and $s'$ the individual strategy of the selfish player $j$.

As in the homogeneous case, the player earns at most $T_{\text{init}} + \#\text{Comm}$ during both initialization and communication. She can indeed choose her rank at the end of the initialization, but this has no impact on the remaining of the algorithm (except for a $M^3K$ term due to the length of the last uncompleted superblock), thanks to Lemma 25.

With probability $1-\mathcal{O}\left(\frac{KM+M\log(T)}{T}\right)$, the initialization is successful, the arms are correctly estimated and no punishment protocol is due to unfortunate inspections (as already explained in Section D.2). The following is conditioned on this event.

Proposition 2 holds independently of the strategy of the selfish player. Moreover, the exploiting players run the RSD algorithm only between the exploiters. This means that when all cooperative players are exploiting, if the selfish player did not signal her preferences, she would always be the last dictator in the RSD algorithm. Because of this, it is in her interest to report as soon as possible her preferences.

Moreover, reporting truthfully is a dominant strategy for the RSD algorithm, meaning that when all players are exploiting, the expected utility received by the selfish player is at most the utility she would get by reporting truthfully. As a consequence, the selfish player can improve her expected reward by at most the length of a superblock during the exploitation phase. Wrapping up all of this and defining $t_0$ the time at which all other players start exploiting:

$$\mathbb{E}\left[\mathrm{Rew}^j_{T_{\text{punish}}+t_p}(s', s_{-j})\right] \leq t_0 + (T_{\text{punish}} + t_p - t_0)\mathbb{E}_{\sigma\sim\mathcal{U}(\mathfrak{S}_M)}\left[\mu^j_{\pi_\sigma(\sigma^{-1}(j))}\right] + \mathcal{O}(M^3K).$$

with $t_0 = \mathcal{O}\left(\frac{K\log(T)}{\Delta^2} + K^2\log(T)\right)$. Lemma 24 then yields for $\widetilde{\alpha} = \frac{1+\left(\frac{1+\delta}{1-\delta}\right)^2\alpha}{2}$:

$$\mathbb{E}\left[\mathrm{Rew}^j_T(s', s_{-j})\right] \leq t_0+(T_{\text{punish}}+t_p-t_0)\mathbb{E}_{\sigma\sim\mathcal{U}(\mathfrak{S}_M)}\left[\mu^j_{\pi_\sigma(\sigma^{-1}(j))}\right]+\widetilde{\alpha}\frac{T - T_{\text{punish}} - t_p}{M}\sum_{k=1}^{M}\mu^j_{(k)}+\mathcal{O}(M^3K).$$

Thanks to Lemma 23, $\mathbb{E}_{\sigma\sim\mathcal{U}(\mathfrak{S}_M)}\left[\mu^j_{\pi_\sigma(\sigma^{-1}(j))}\right] \geq \frac{\sum_{k=1}^M \mu^j_{(k)}}{M}$. We assume $\delta < \frac{1-(1-1/K)^{\frac{M-1}{2}}}{1+(1-1/K)^{\frac{M-1}{2}}}$ here, so that $\widetilde{\alpha} < 1$. Because of this, the right term is maximized when $T_{\text{punish}}$ is maximized, i.e., equal to $T$. Then:

$$\mathbb{E}\left[\mathrm{Rew}^j_T(s', s_{-j})\right] \leq T\mathbb{E}_{\sigma\sim\mathcal{U}(\mathfrak{S}_M)}\left[\mu^j_{\pi_\sigma(\sigma^{-1}(j))}\right] + t_0 + t_p + \mathcal{O}(M^3K).$$

Using the first point of Theorem 7 to compare $T\mathbb{E}_{\sigma\sim\mathcal{U}(\mathfrak{S}_M)}\left[\mu^j_{\pi_\sigma(\sigma^{-1}(j))}\right]$ with $\mathrm{Rew}^j_T(s)$ and adding the low probability event then yields the first point of Theorem 7. ■

**Proof of the second point of Theorem 7 (stability).** For $p_0 = \mathcal{O}\left(\frac{KM+M\log(T)}{T}\right)$, with probability at least $1-p_0$, the initialization is successful, the cooperative players start exploiting with correct estimated preferences after a time at most $t_0 = \mathcal{O}\left(K^2\log(T) + \frac{K\log(T)}{\Delta^2}\right)$ and no punishment protocol is started due to unfortunate inspections. Define $\varepsilon' = t_0 + Tp_0 + 7M^3K$. Assume that the player $j$ is playing a deviation strategy $s'$ such that for some $i$ and $l > 0$:

$$\mathbb{E}\left[\mathrm{Rew}^i_T(s', s_{-j})\right] \leq \mathbb{E}\left[\mathrm{Rew}^i_T(s)\right] - l - \varepsilon'$$

First, let us fix $\sigma$ the permutation returned by the initialization, $T_{\text{punish}}$ the time at which the punishment protocol starts and divide $l = l_{\text{before punishment}} + l_{\text{after punishment}}$ in two terms: the regret incurred before the punishment protocol and the regret after. Let us now compare $s'$ with $s^*$, the optimal strategy for player $j$. Let $\varepsilon$ take account of the low probability event of a bad initialization/exploration, the last superblock that remains uncompleted, the time before all cooperative players start the exploitation and the event that a punishment accidentally starts. Thus the only way for player $i$ to suffer some additional regret before punishment is to lose it during a completed superblock of the exploitation. Three cases are possible:

1. The selfish player truthfully reports her preferences. The average utility of player $i$ during the exploitation is then $\widetilde{U}_i(\sigma)$ as defined in Lemma 25. The only way to incur some additional loss to player $i$ before the punishment is then to collide with her, in which case her loss is at most $(1+\delta)\mu_{(1)}$ while the selfish player's loss is at least $(1-\delta)\mu_{(M)}$.

After $T_{\text{punish}}$, Lemma 24 yields that the selfish player suffers a loss at least $(1-\widetilde{\alpha})\frac{T-T_{\text{punish}}-t_p}{M}\sum_{k=1}^M \mu^j_{(k)}$, while any cooperative player $i$ suffers a loss at most $(T - T_{\text{punish}})\widetilde{U}_i(\sigma)$. Thanks to Lemma 23 and the $\delta$-heterogeneity assumption, this term is smaller than $\frac{T-T_{punish}}{M}\left(\frac{1+\delta}{1-\delta}\right)^3\sum_{k=1}^M \mu^j_{(k)}$.

Then, the selfish player after $T_{\text{punish}}$ suffers a loss at least $\frac{(1-\widetilde{\alpha})(1-\delta)^3}{(1+\delta)^3} l_{\text{after punish}} - t_p$.

In the first case, we thus have for $\beta = \min(\frac{(1-\widetilde{\alpha})(1-\delta)^3}{(1+\delta)^3}, \frac{(1-\delta)\mu_{(M)}}{(1+\delta)\mu_{(1)}})$:

$$\mathbb{E}[\text{Rew}_T^j(s', s_{-j})|\sigma] \leq \mathbb{E}[\text{Rew}_T^j(s^*, s_{-j})|\sigma] - \beta l + t_p.$$

2. The selfish player never reports her preferences. In this case, it is obvious that the utility returned by the assignments to any other player is better than if the selfish player reports truthfully. Then the only way to incur some additional loss to player $i$ before punishment is to collide with her, still leading to a ratio of loss at most $\frac{\mu^j_{(M)}}{\mu^i_{(1)}}$.

From there, it can be concluded as in the first case that for $\beta = \min(\frac{(1-\widetilde{\alpha})(1-\delta)^3}{(1+\delta)^3}, \frac{(1-\delta)\mu_{(M)}}{(1+\delta)\mu_{(1)}})$:

$$\mathbb{E}[\text{Rew}_T^j(s', s_{-j})|\sigma] \leq \mathbb{E}[\text{Rew}_T^j(s^*, s_{-j})|\sigma] - \beta l + t_p.$$

3. The selfish player reported fake preferences. If these fake preferences never change the issue of the `ComputeRSD` protocol, this does not change from the first case. Otherwise, for any block where the final assignment is changed, the selfish player does not receive the arm she would get if she reported truthfully. Denote by $n$ the number of such blocks, by $N_{\text{lie}}$ the number of times player $j$ did not pull the arm attributed by `ComputeRSD` during such a block before $T_{\text{punish}}$ and by $l_b$ the loss incurred to player $i$ on the other blocks.

As for the previous cases, the loss incurred by the selfish player during the blocks where the assignment of `ComputeRSD` is unchanged is at least $\frac{(1-\delta)\mu_{(M)}}{(1+\delta)\mu_{(1)}} l_b$.

Each time the selfish player pulls the attributed arm by `ComputeRSD` in a block where the assignment is changed, she suffers a loss at least $\Delta$. The total loss for the selfish player is then (w.r.t. the optimal strategy $s^*$) at least:

$$(1-\widetilde{\alpha})\frac{T - T_{\text{punish}} - t_p}{M} \sum_{k=1}^{M} \mu^j_{(k)} + \left(\frac{n}{M}(T_{\text{punish}} - t_0) - N_{\text{lie}}\right)\Delta + \frac{(1-\delta)\mu_{(M)}}{(1+\delta)\mu_{(1)}} l_b.$$

On the other hand, the loss for a cooperative player is at most:

$$\frac{T - T_{\text{punish}}}{M}\left(\frac{1+\delta}{1-\delta}\right)^3 \sum_{k=1}^{M} \mu^j_{(k)} + \frac{n}{M}(T_{\text{punish}} - t_0)(1+\delta)\mu_{(1)} + l_b.$$

Moreover, each time the selfish player does not pull the attributed arm by `ComputeRSD`, she has a probability $\widetilde{p} = 1 - (1 - \frac{p}{M-1})^{M-1} \geq \frac{p}{2}$ for $p = \frac{\sqrt{\log(T)}}{T}$, to receive a random inspection and thus to trigger the punishment protocol. Because of this, $N_{\text{lie}}$ follows a geometric distribution of parameter $\widetilde{p}$ and $\mathbb{E}[N_{\text{lie}}] \leq \frac{2}{p}$.

When taking the expectations over $T_{\text{punish}}$ and $N_{\text{lie}}$, but still fixing $\sigma$ and $n$, we get:

$$l_{\text{selfish}} \geq (1-\widetilde{\alpha})\frac{T - \mathbb{E}[T_{\text{punish}}] - t_p}{M}\sum_{k=1}^{M}\mu_{(k)}^{j} + \left(\frac{n}{M}\left(\mathbb{E}[T_{\text{punish}}] - t_0\right) - 2/p\right)\Delta + \frac{(1-\delta)\mu_{(M)}}{(1+\delta)\mu_{(1)}}l_b,$$
$$l \leq \frac{T - \mathbb{E}[T_{\text{punish}}]}{M}\left(\frac{1+\delta}{1-\delta}\right)^3\sum_{k=1}^{M}\mu_{(k)}^{j} + \frac{n}{M}(\mathbb{E}[T_{\text{punish}}] - t_0)(1+\delta)\mu_{(1)} + l_b.$$

First assume that $\frac{n}{M}(\mathbb{E}[T_{\text{punish}}] - t_0) \geq \frac{4}{p}$. In that case, we get:

$$l_{\text{selfish}} \geq (1-\widetilde{\alpha})\frac{T - \mathbb{E}[T_{\text{punish}}] - t_p}{M}\sum_{k=1}^{M}\mu_{(k)}^{j} + \frac{n}{2M}(\mathbb{E}[T_{\text{punish}}] - t_0)\Delta + \frac{(1-\delta)\mu_{(M)}}{(1+\delta)\mu_{(1)}}l_b,$$
$$l \leq \frac{T - \mathbb{E}[T_{\text{punish}}]}{M}\left(\frac{1+\delta}{1-\delta}\right)^3\sum_{k=1}^{M}\mu_{(k)}^{j} + \frac{n}{M}(\mathbb{E}[T_{\text{punish}}] - t_0)(1+\delta)\mu_{(1)} + l_b.$$

In the other case, we have by noting that $(1+\delta)\mu_{(1)} \leq \frac{1+\delta}{1-\delta}\sum_{k=1}^{M}\mu_{(k)}^{j}$:

$$l_{\text{selfish}} \geq (1-\widetilde{\alpha})T\left(1 - \frac{4M}{\sqrt{\log(T)}} - t_p\right)\frac{1}{M}\sum_{k=1}^{M}\mu_{(k)}^{j} + \frac{(1-\delta)\mu_{(M)}}{(1+\delta)\mu_{(1)}}l_b,$$
$$l \leq T\left(1 + \frac{4M}{\sqrt{\log(T)}}\right)\frac{1}{M}\left(\frac{1+\delta}{1-\delta}\right)^3\sum_{k=1}^{M}\mu_{(k)}^{j} + l_b.$$

In any of these two cases, for $\widetilde{\beta} = \min\left((1-\widetilde{\alpha})\left(\frac{1+\delta}{1-\delta}\right)^3\frac{\sqrt{\log(T)}-4M}{\sqrt{\log(T)}+4M}; \frac{\Delta}{(1+\delta)\mu_{(1)}}; \frac{(1-\delta)\mu_{(M)}}{(1+\delta)\mu_{(1)}}\right)$:

$$l_{\text{selfish}} \geq \widetilde{\beta}l - t_p$$

Let us now gather all the cases. When taking the previous results in expectation over $\sigma$, this yields for the previous definition of $\widetilde{\beta}$:

$$\mathbb{E}[\text{Rew}_T^i(s', s_{-j})] \leq \mathbb{E}[\text{Rew}_T^i(s)] - l - \varepsilon' \implies \mathbb{E}[\text{Rew}_T^j(s', s_{-j})] \leq \mathbb{E}[\text{Rew}_T^j(s^*, s_{-j})] - \widetilde{\beta}l + t_p + t_0.$$

Moreover, thanks to the second part of Theorem 7, $\mathbb{E}[\text{Rew}_T^j(s^*, s_{-j})] \leq \mathbb{E}[\text{Rew}_T^j(s)] + \varepsilon$, with $\varepsilon = \mathcal{O}\left(\frac{K\log(T)}{\Delta^2} + K^2\log(T) + \frac{K\log(T)}{(1-\delta)r^2\mu_{(K)}}\right)$. Then by defining $l_1 = l + \varepsilon'$, $\varepsilon_1 = \varepsilon + t_p + t_0 + \widetilde{\beta}\varepsilon' = \mathcal{O}(\varepsilon)$, we get:

$$\mathbb{E}[\text{Rew}_T^i(s', s_{-j})] \leq \mathbb{E}[\text{Rew}_T^i(s)] - l_1 \implies \mathbb{E}[\text{Rew}_T^j(s', s_{-j})] \leq \mathbb{E}[\text{Rew}_T^j(s)] - \widetilde{\beta}l_1 + \varepsilon_1.$$

■

### D.3.1. AUXILIARY LEMMAS

**Proof of Lemma 23.** Assume that player $j$ is the $k$-th dictator for an RSD assignment. Since only $k-1$ arms are reserved before she chooses, she earns at least $\mu^j_{(k)}$ after this assignment. This yields the first inequality:

$$\widetilde{U}_j(\sigma) \geq \frac{\sum_{k=1}^M \mu^j_{(k)}}{M}$$

Still assuming that player $j$ is the $k$-th dictator, let us prove that she earns at most $\left(\frac{1+\delta}{1-\delta}\right)^2 \mu^j_{(k)}$. Assume w.l.o.g. that she ends up with the arm $l$ such that $\mu^j_l > \mu^j_{(k)}$. This means that a dictator $j'$ before her preferred an arm $i$ to the arm $l$ with $\mu^j_l > \mu^j_{(k)} \geq \mu^j_i$.

Since $j'$ preferred $i$ to $l$, $\mu^{j'}_i \geq \mu^{j'}_l$. Using the $\delta$-heterogeneity assumption, it comes:

$$\mu^j_l \leq \frac{1+\delta}{1-\delta}\mu^{j'}_l \leq \frac{1+\delta}{1-\delta}\mu^{j'}_i \leq \left(\frac{1+\delta}{1-\delta}\right)^2 \mu^j_i \leq \left(\frac{1+\delta}{1-\delta}\right)^2 \mu^j_{(k)}$$

Thus, player $j$ earns at most $\left(\frac{1+\delta}{1-\delta}\right)^2 \mu^j_{(k)}$ after this assignment, which yields the second inequality of Lemma 23.

■

**Proof of Lemma 24.** The punishment protocol starts for all cooperative players at $T_{\text{punish}}$. Define $\alpha' = \left(\frac{1+\delta}{1-\delta}\right)^2 \gamma$ and $\delta' = \frac{1-\alpha'}{1+3\alpha'}$. The condition $r > 0$ is equivalent to $\delta' > 0$.

As in the homogeneous case, each player then estimates each arm such that after $t_p = \mathcal{O}\left(\frac{K\log(T)}{(1-\delta)\cdot(\delta')^2\mu_{(K)}}\right)$[10] rounds, $(1-\delta')\widehat{\mu}^j_k \leq \mu^j_k \leq (1+\delta)\widehat{\mu}^j_k$ with probability $1-\mathcal{O}\left(KM/T\right)$, thanks to Lemma 22. This implies that for any cooperative player $j'$:

$$\begin{aligned}
1 - p^{j'}_k &\leq \left(\gamma\frac{(1+\delta')\sum_{m=1}^M \mu^{j'}_{(m)}}{(1-\delta')M\mu^{j'}_k}\right)^{\frac{1}{M-1}} \\
&\leq \left(\gamma\frac{1+\delta'}{1-\delta'}\left(\frac{1+\delta}{1-\delta}\right)^2 \frac{\sum_{m=1}^M \mu^j_{(m)}}{M\mu^j_k}\right)^{\frac{1}{M-1}}
\end{aligned}$$

The last inequality is due to the fact that in the $\delta$-heterogeneous setting, $\frac{\mu^j_k}{\mu^{j'}_k} \in [\left(\frac{1-\delta}{1+\delta}\right)^2, \left(\frac{1+\delta}{1-\delta}\right)^2]$. Thus, the expected reward that gets the selfish player $j$ by pulling $k$ after the time $T_{\text{punish}} + t_p$ is smaller than $\gamma\frac{1+\delta'}{1-\delta'}\left(\frac{1+\delta}{1-\delta}\right)^2 \frac{\sum_{m=1}^M \mu^j_{(m)}}{M}$.

Note that $\gamma\frac{1+\delta'}{1-\delta'}\left(\frac{1+\delta}{1-\delta}\right)^2 = \widetilde{\alpha}$. Considering the low probability event of bad estimations of the arms adds a constant term that can be counted in $t_p$, leading to Lemma 24. ■

10. The $\delta$-heterogeneous assumption is here used to say that $\frac{1}{\mu^j_{(K)}} \leq \frac{1}{(1-\delta)\mu_{(K)}}$.

**Proof of Lemma 25.** Consider the selfish player $j$ and denote $\sigma$ the permutation given by the initialization. The rank of player $j'$ is then $\sigma^{-1}(j')$. All other players $j$ pull uniformly at random until having an attributed rank. Moreover, player $j$ does not know the players with which she collides. This implies that she can not correlate her rank with the rank of a specific player, i.e., $\mathbb{P}_\sigma\left[\sigma(k') = j' | \sigma(k) = j\right]$ does not depend on $j'$ as long as $j' \neq j$.

This directly implies that the distribution of $\sigma | \sigma(k) = j$ is uniform over $\mathfrak{S}_M^{j \to k}$. Thus, the distribution of $\sigma \circ \sigma_0^{-l} | \sigma(k) = j$ is uniform over $\mathfrak{S}_M^{j \to k+l \pmod M}$ and finally for any $j' \in [M]$:

$$\begin{aligned}
\mathbb{E}_{\sigma \sim \text{successful initialization}} \left[ \frac{1}{M} \sum_{l=1}^{M} \mu^j_{\pi_{\sigma \circ \sigma_0^{-l}}\left(\sigma_0^l \circ \sigma^{-1}(j)\right)} \,\middle|\, \sigma(k) = j \right] &= \frac{1}{M} \sum_{l=1}^{M} \mathbb{E}_{\sigma \sim \mathcal{U}\left(\mathfrak{S}_M^{j \to l}\right)} \left[ \mu^{j'}_{\pi_\sigma(\sigma^{-1}(j'))} \right], \\
&= \frac{1}{M} \sum_{l=1}^{M} \frac{1}{(M-1)!} \sum_{\sigma \in \mathfrak{S}_M^{j \to l}} \mu^{j'}_{\pi_\sigma(\sigma^{-1}(j'))}, \\
&= \frac{1}{M!} \sum_{\sigma \in \mathfrak{S}_M} \mu^{j'}_{\pi_\sigma(\sigma^{-1}(j'))}.
\end{aligned}$$

Taking the expectation of the left term then yields Lemma 25. ■