ECAI 2020
G.D. Giacomo et al. (Eds.)

doi:10.3233/FAIA200081

Parameterized Complexity of Manipulating Sequential Allocation

Michele Flammini and **Hugo Gilbert** [1]

Abstract. The sequential allocation protocol is a simple and popular mechanism to allocate indivisible goods, in which the agents take turns to pick the items according to a predefined sequence. While this protocol is not strategy-proof, it has been recently shown that finding a successful manipulation for an agent is an NP-hard problem [1]. Conversely, it is also known that finding an optimal manipulation can be solved in polynomial time in a few cases: if there are only two agents or if the manipulator has a binary or a lexicographic utility function. In this work, we take a parameterized approach to provide several new complexity results on this manipulation problem. More precisely, we give a complete picture of its parameterized complexity w.r.t. the following three parameters: the number n of agents, the number $\mu(a_1)$ of times the manipulator a_1 picks in the picking sequence, and the maximum range $\texttt{rg}^{\max}$ of an item. This third parameter is a correlation measure on the preference rankings of the agents. In particular, we provide XP algorithms for parameters n and $\mu(a_1)$, and we show that the problem is fixed-parameter tractable w.r.t. $\texttt{rg}^{\max}$ and $n + \mu(a_1)$. Interestingly enough, we show that w.r.t. the single parameters n and $\mu(a_1)$ it is W[1]-hard.

1 INTRODUCTION

Allocating resources to a set of agents in an efficient and fair manner is one of the most fundamental problems in computational social choice. One challenging case is the allocation of indivisible items [5, 8, 17], e.g., allocating players to teams. To address this problem, the sequential allocation mechanism has lately received increasing attention in the AI literature [3, 4, 6, 14, 15, 16, 20]. This mechanism works as follows: at each time step, an agent, selected according to a predefined sequence, is allowed to pick one item among the remaining ones. Such a protocol has many desirable qualities: it is simple, it can be run both in a centralized and in a decentralized way, and agents do not have to submit cardinal utilities. For these reasons, sequential allocation is used in several real life applications, as for instance by several professional sports associations [9] to organize their draft systems (e.g., the annual draft of the National Basketball Association in the US), and by the Harvard Business School to allocate courses to students [10].

Unfortunately, it is well known that the sequential allocation protocol is not strategy-proof. Stated otherwise, an agent can obtain a better allocation by not obeying her preferences [6]. Knowing that this protocol is manipulable can make its outcome less legitimate: an agent which is unhappy with the result could have the feeling that this is because someone has cheated by manipulating the allocation process. Put in another way, the agents confidence or appreciation of the allocation system decreases if they know that someone can cheat. Such a drawback has motivated the algorithmic study of several issues related to strategic behaviors in the sequential allocation setting, the most important one being the computation of a "successful" manipulation. Notably, Aziz et al. [1] have shown that the problem of manipulating sequential allocation is NP-hard. This hardness result could be seen as an answer to the concern that the sequential allocation protocol is not strategy-proof. Indeed, if finding a successful manipulation is computationally too difficult, then agents may be inclined to behave truthfully [19]. However, this is only a worst-case result which says little about which instances are easy to manipulate and which are not. To provide more information on this question, we take a parameterized complexity approach.

[1] Gran Sasso Science Institute, Italy, email: firstname.lastname@gssi.it

Our contribution. We tackle the parameterized complexity of manipulating sequential allocations, and provide a complete picture of the problem w.r.t. the following three parameters: the number n of agents, the number $\mu(a_1)$ of items the manipulator gets to pick in the allocation process, and the maximum range $\texttt{rg}^{\max}$ of an item, a parameter measuring how close the preference rankings of the agents are. In particular, we provide a novel dynamic programming algorithm, which we show is XP w.r.t. n, and Fixed-Parameter Tractable (FPT) w.r.t. $\texttt{rg}^{\max}$ and the sum $n + \mu(a_1)$. Moreover, we show that the problem is in XP w.r.t. $\mu(a_1)$. Interestingly enough, we finally prove that the problem is W[1]-hard w.r.t. to the single parameters n and $\mu(a_1)$. As a consequence, our XP results are both tight. Table 1 summarizes our results.

Table 1. Our parameterized complexity results on the problem of manipulating sequential allocations.

Parameter	n	$\mu(a_1)$	$n + \mu(a_1)$	$\texttt{rg}^{\max}$
Results	In XP but W[1]-hard Theorems 1 and 6	In XP but W[1]-hard Theorem 5	In FPT Theorem 2	In FPT Theorem 4

Related work on strategic behaviors in the sequential allocation setting. Aziz et al. [2] studied the sequential allocation setting by treating it as a one shot game. They notably designed a linear-time algorithm to compute a pure Nash equilibrium and a polynomial-time algorithm to compute an optimal Stackelberg strategy when there are two agents, a leader and a follower, and the follower has a constant number of distinct values for the items. If the sequential allocation setting is seen as a finite repeated game with perfect information, Kalinowski et al. [15] showed that the unique subgame perfect Nash equilibrium can be computed in linear time when there are two players. However, with more agents, the authors showed that computing one of the possibly exponentially many equilibria is PSPACE-hard.

Several papers focused on the complexity of finding a successful manipulation for a given agent, also called manipulator. Bouveret and Lang [6] showed that determining if there exists a strategy for the manipulator to get a specific set of items can be done in polynomial time. Moreover, Aziz et al. [1] showed that finding if there exists a successful manipulation, whatever the utility function of the manipulator, is a polynomial-time problem. Conversely, the same authors showed that determining an optimal manipulation (for a specific utility function) is an NP-hard problem. Bouveret and Lang [7] provided further hardness results for the cases of non-additive preferences and coalitions of manipulators. On the other hand, finding an optimal manipulation can be performed in polynomial time if the manipulator has lexicographic or binary utilities [1, 7] or if there are only two agents [7]. Tominaga et al. [18] further showed that finding an optimal manipulation is a polynomial-time problem when there are two agents and the picking sequence is composed of a sequence of randomly generated rounds. More precisely, in each round, both agents get to pick one item and a coin flip determines who picks first. Lastly, an XP algorithm for the number of agents has been recently and independently designed by Xiao and Ling [20], showing that manipulating sequential allocations can be performed in polynomial time when the number of agents is bounded. Interestingly, we provide a simpler algorithm with the same property and which we analyse to obtain further positive parameterized complexity results.

2 SETTINGS AND NOTATIONS

We consider a set $\mathcal{A} = \{a_1, \dots, a_n\}$ of n agents and a set $\mathcal{I} = \{i_1, \dots, i_m\}$ of m items. A preference profile $P = \{\succ_{a_1}, \dots, \succ_{a_n}\}$ describes the preferences of the agents. More precisely, P is a collection of rankings such that ranking $\succ_a$ specifies the preferences of agent a over the items in $\mathcal{I}$. The items are allocated to the agents according to the following sequential allocation procedure: at each time step, a picking sequence $\pi \in \mathcal{A}^m$ specifies an agent who gets to pick an item within the remaining ones. Put another way, $\pi(1)$ picks first, then $\pi(2)$ picks second, and so forth. We assume that agents behave greedily by choosing at each time step their preferred item within the remaining ones. If we view sequential allocation as a centralized protocol, then all agents report their preference rankings to a central authority which mimics this picking process. In the following, w.l.o.g. we use this centralized viewpoint where agents have to report their preference rankings. This sequential process leads to an allocation that we denote by ϕ. More formally, ϕ is a function such that $\phi(a)$ is the set of items that agent a has obtained at the end of the sequential allocation process.

Example 1 (Adapted from Example 1 in [1])**.** *For the sake of illustration, we consider an instance with 3 agents and 4 items, i.e., $\mathcal{A} = \{a_1, a_2, a_3\}$ and $\mathcal{I} = \{i_1, i_2, i_3, i_4\}$. The preferences of the agents are described by the following profile:*

$$a_1 : i_1 \succ i_2 \succ i_3 \succ i_4 \qquad a_2 : i_3 \succ i_4 \succ i_1 \succ i_2$$
$$a_3 : i_1 \succ i_2 \succ i_3 \succ i_4$$

and the picking sequence is $\pi = (a_1, a_2, a_3, a_1)$. Then, a_1 will first pick i_1, then a_2 will pick i_3, a_3 will pick i_2, and lastly a_1 will pick i_4. Hence, the resulting allocation is given by $\phi(a_1) = \{i_1, i_4\}$, $\phi(a_2) = \{i_3\}$ and $\phi(a_3) = \{i_2\}$.

The allocation ϕ is completely determined by the picking sequence π and the preference profile P. Notably, if one of the agents reports a different preference ranking, she may obtain a different set of items. Such a set may even be more desirable to her. Consequently, agents may have an incentive to misreport their preferences.

Example 2 (Example 1 continued)**.** *Assume now that agent a_1 reports the preference ranking $i_3 \succ i_2 \succ i_1 \succ i_4$. Then she obtains the set $\phi(a_1) = \{i_2, i_3\}$. This set may be more desirable to a_1 than $\{i_1, i_4\}$ if for instance items i_1, i_2 and i_3 are almost as desirable as one another, but all the three are much more desirable than i_4.*

In this work, we study this type of manipulation. We will assume that agent a_1 is the manipulator and that all other agents behave truthfully. Although the agents are asked to report ordinal preferences, we will assume a standard assumption in the literature that a_1 has underlying additive utilities for the items. More formally, the preferences of a_1 over items in $\mathcal{I}$ are described by a set of positive values $U = \{u(i) | i \in \mathcal{I}\}$ such that $i \succ_{a_1} j$ implies $u(i) > u(j)$. The utility of a set of items S is then obtained by summation, i.e., $u(S) = \sum_{i \in S} u(i)$. We will denote by $\succ_T$ the truthful preference ranking of a_1, and by $\phi_\succ$ the allocation obtained if agent a_1 reports the preference ranking $\succ$. Moreover, we will denote by u_T the value $u(\phi_{\succ_T}(a_1))$, which is the utility value of a_1's allocation when she behaves truthfully. We will say that a preference ranking $\succ$ is a successful manipulation if a_1 prefers $\phi_\succ(a_1)$ to $\phi_{\succ_T}(a_1)$, i.e., if $u(\phi_\succ(a_1)) > u_T$. Hence, the objective for the manipulator is to find a successful manipulation $\succ$ maximizing $u(\phi_\succ(a_1))$. We are now ready to define formally the problem of **Manip**ulating a **S**equential **A**llocation process, called $ManipSA$.

$ManipSA$
Input: A set $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$ of n agents where a_1 is the manipulator, a set $\mathcal{I} = \{i_1, \dots, i_m\}$ of m items, a picking sequence π, a preference profile P and a set U of utility values for a_1.
Find: A preference ranking $\succ$ maximizing $u(\phi_\succ(a_1))$.

The $ManipSA$ problem is known to be NP-hard [1]. In this work, we address this optimization problem from a parameterized complexity point of view. We will be mostly interested in three types of parameters: the number of agents, the number of items that an agent gets to pick, and the range of the items. While the number n of agents is already clear, let us define the other parameters more formally. We denote by $\mu(a)$ the number of items that agent a gets to choose in π and by $\mu^{\max} = \max\{\mu(a) | a \in \mathcal{A}\}$ the maximum of these values. Let $\mathtt{rk}_a(i)$ denote the rank of item i in the preference ranking of agent a. Then, we define the range $\mathtt{rg}(i)$ of an item i as:

$$\mathtt{rg}(i) = \max_{a \in \mathcal{A} \setminus \{a_1\}} \mathtt{rk}_a(i) - \min_{a \in \mathcal{A} \setminus \{a_1\}} \mathtt{rk}_a(i) + 1.$$

Note that $\mathtt{rg}(i)$ is defined using only non-manipulators. The maximum range of an item $\mathtt{rg}^{\max}$ is then defined as $\max_{i \in \mathcal{I}} \mathtt{rg}(i)$.

Let us give some intuitions on parameters $\mu(a_1)$ and $\mathtt{rg}^{\max}$. In the $ManipSA$ problem, $\mu(a_1)$ can be seen as a budget parameter for the manipulator. Intuitively, the larger the value of $\mu(a_1)$, the more she can manipulate. It is also the size of the bundle a_1 will get. Interestingly, in real-life applications, $\mu(a_1)$ can be much smaller than $|\mathcal{I}|$ and even much smaller than $|\mathcal{A}|$. For these reasons, $\mu(a_1)$ is an interesting parameter to study in the $ManipSA$ problem. On the other hand, parameter $\mathtt{rg}^{\max}$ measures the correlation between the preferences of the non-manipulators. If $\mathtt{rg}^{\max} = 1$ (its minimal possible value), then all non-manipulators have the same preference ranking. In this case, the manipulation problem becomes easy to solve, as all non-manipulators can be treated as a single agent and the case of two

agents is known to be polynomial-time solvable. This simple insight can let us hope that the manipulation problem remains tractable if this parameter is small. An important motivation behind parameter $\mathtt{rg}^{\max}$ is that, in practice, the preferences of different agents are often correlated.

3 POSITIVE PARAMETERIZED COMPLEXITY RESULTS

To solve the $ManipSA$ problem, one can simply try the $m!$ possible preference rankings and see which ones yield the maximum utility value. However, a more clever approach uses the following result.

Fact 1 (From Propositions 7 and 8 in [6])**.** *Given a specific set of items S, it is possible to determine in polynomial time whether there exists a preference ranking $\succ$ such that $S \subseteq \phi_{\succ}(a_1)$. In such case, computing such a ranking can also be done in polynomial time.*

Hence, one can try all the $\binom{m}{\mu(a_1)}$ possible sets of $\mu(a_1)$ items to determine which is the best one that a_1 can get. This approach shows that $ManipSA$ is in XP w.r.t. parameter $\mu(a_1)$.

Example 3 (Example 1 continued)**.** *As in this example $\mu(a_1) = 2$, a_1 just has to determine which one of the $\binom{4}{2}$ following sets she must obtain: $\{i_1, i_2\}, \{i_1, i_3\}, \{i_1, i_4\}, \{i_2, i_3\}, \{i_2, i_4\}, \{i_3, i_4\}$.*

To obtain further positive results, we first design a dynamic programming method. We will then explain how it entails several positive parameterized complexity results.

3.1 A dynamic programming algorithm

Our dynamic programming approach considers pairs (k, S) where $S \subseteq \mathcal{I}$ and $k \in \{0, 1, \ldots, \mu(a_1)\}$. In a state characterized by the pair (k, S), we know that the items in S have been picked by some agents in $\mathcal{A}$ as well as k other items that have been picked by agent a_1. However, while the items in S are clearly identified, the identities of these k other items are unspecified.

Given a pair (k, S), the number of items that have already been picked is $|S| + k$. Hence, the next picker is $\pi(|S| + k + 1)$. Let us denote this agent by a. If $a = a_1$, i.e., she is the manipulator, then she will pick one more item within the set $\mathcal{I} \setminus S$ and we move to state $(k + 1, S)$. Otherwise, let $b(a, S)$ denote the preferred item of agent a within the set $\mathcal{I} \setminus S$. Then, two cases are possible:

- In the first case, $b(a, S)$ has already been picked by agent a_1. Then, we move to state $(k - 1, S \cup \{b(a, S)\})$. Note that this is only possible if k was greater than or equal to one.
- Otherwise $b(a, S)$ is picked by agent a and we move to state $(k, S \cup \{b(a, S)\})$.

Let us denote by $V(k, S)$ the maximal utility that the manipulator can get from items in $\mathcal{I} \setminus S$ starting from state (k, S). Then the value of an optimal manipulation is given by $V(k = 0, S = \emptyset)$. From the previous analysis, function V verifies the following dynamic programming equations:

$$V(k, S) = V(k + 1, S) \text{ if } \pi(|S| + k + 1) = a_1 \quad (1)$$

$$V(k = 0, S) = V(k, S \cup \{b(a, S)\}) \text{ if } \pi(|S| + k + 1) = a \neq a_1 \quad (2)$$

$$V(k > 0, S) = \max\big(V(k - 1, S \cup \{b(a, S)\}) + u(b(a, S)), V(k, S \cup \{b(a, S)\})\big) \text{ if } \pi(|S| + k + 1) = a \neq a_1 \quad (3)$$

where the termination is guaranteed by the fact that $V(k, S) = \sum_{i \in \mathcal{I} \setminus S} u(i)$ when $|S| + k = m$.

Let us detail, once again, Equation 3. In this case, the current picker a is not the manipulator. As we know that items in S have previously been picked, the best possibly available item for her is $b(a, S)$. However, $b(a, S)$ can only be picked by a if it is not one of the unknown $k > 0$ items previously picked by a_1. Hence, $V(k, S)$ is the maximum of two values corresponding to two possible cases. Either a_1 has previously picked $b(a, S)$, in which case we should count the utility $u(b(a, S))$ of this item as well as the value $V(k - 1, S \cup \{b(a, S)\})$ of the corresponding next state; or $b(a, S)$ can be picked by agent a and we consider the value $V(k, S \cup \{b(a, S)\})$ of the corresponding next state.

Equations 1-3 induce a directed acyclic state graph $G_{\mathtt{dp}} = (V_{\mathtt{dp}}, A_{\mathtt{dp}})$, where $V_{\mathtt{dp}}$ is the set of states generated from state $(k = 0, S = \emptyset)$ when using these equations and the arcs in $A_{\mathtt{dp}}$ connect each state to its successor states. We will also denote by $\mathcal{S}_{\mathtt{dp}} = \{S | (k, S) \in V_{\mathtt{dp}}\}$ the set of item-sets S that are involved in $V_{\mathtt{dp}}$.

Notably, solving Equations 1-3 can be performed by building graph $G_{\mathtt{dp}}$ and running backward induction on it (from the lastly generated states to the initial state $(k = 0, S = \emptyset)$). By standard bookkeeping techniques one can also identify the items that are taken in an optimal manipulation and the order in which they are taken and then recover an optimal ranking to report (where these items are ranked first and in the same order).

Example 4 (Example 1 continued)**.** *Let us illustrate our approach on our running example. We let $u(i_1) = 5$, $u(i_2) = 4$, $u(i_3) = 3$ and $u(i_4) = 1$. Figure 1 displays the resulting state graph $G_{\mathtt{dp}}$. The values of the states are given next to them and the optimal branches are displayed with thick solid arrows. In this example, $\mathcal{S}_{\mathtt{dp}} = \{\emptyset, \{i_3\}, \{i_1, i_3\}, \{i_3, i_4\}, \{i_1, i_2, i_3\}, \{i_1, i_3, i_4\}\}$. As we can see, the optimal choices for a_1 is to first pick i_3 so that i_2 is still available the second time she becomes the picker. Hence, an optimal manipulation is given by $\succ = i_3 \succ i_2 \succ i_1 \succ i_4$ which results in an allocation $\phi_{\succ}(a_1) = \{i_2, i_3\}$ with a utility of 7 whereas $u_T = 6$.*

3.2 Complexity analysis

We now provide positive parameterized complexity results by proving several upper bounds on $|V_{\mathtt{dp}}|$. In fact, we will prove bounds on $|\mathcal{S}_{\mathtt{dp}}|$ and use the observation that $|V_{\mathtt{dp}}| \leq (\mu(a_1)+1)|\mathcal{S}_{\mathtt{dp}}| \leq m|\mathcal{S}_{\mathtt{dp}}|$ as there are only $\mu(a_1) + 1$ possible values for k in a state (k, S).

The algorithm is XP w.r.t. parameter n. Let $D(a, i)$ denote the set of items that agent a prefers to item i, i.e., $D(a, i) = \{j \in \mathcal{I} | j \succ_a i\}$. Then, for any set $S \subseteq \mathcal{I}$, the definition of $b(a, S)$, which we recall is the preferred element of agent a in set $\mathcal{I} \setminus S$ implies that $\bigcup_{a \in \mathcal{A} \setminus \{a_1\}} D(a, b(a, S)) \subseteq S$. Let us denote by Δ the set of item-sets for which the equality holds, i.e., $\Delta = \{S \subseteq \mathcal{I} | \bigcup_{a \in \mathcal{A} \setminus \{a_1\}} D(a, b(a, S)) = S\}$. Note that a set $S \in \Delta$ is completely determined by the vector $(b(a_2, S), \ldots, b(a_n, S))$ and thus $|\Delta| \leq m^{n-1}$. Our first key insight is that $\mathcal{S}_{\mathtt{dp}}$ is a subset of Δ.

Lemma 1. *The set $\mathcal{S}_{\mathtt{dp}}$ is a subset of Δ.*

Proof. The result follows simply by induction. The result is true for the initial state in which $S = \emptyset$. Assume that the result is true for a state characterized by a pair (k, S). We show that the result also holds for the successor states. If $\pi(|S| + k + 1) = a_1$, then the successor state is $(k + 1, S)$ so the result is also true for this new state as S is

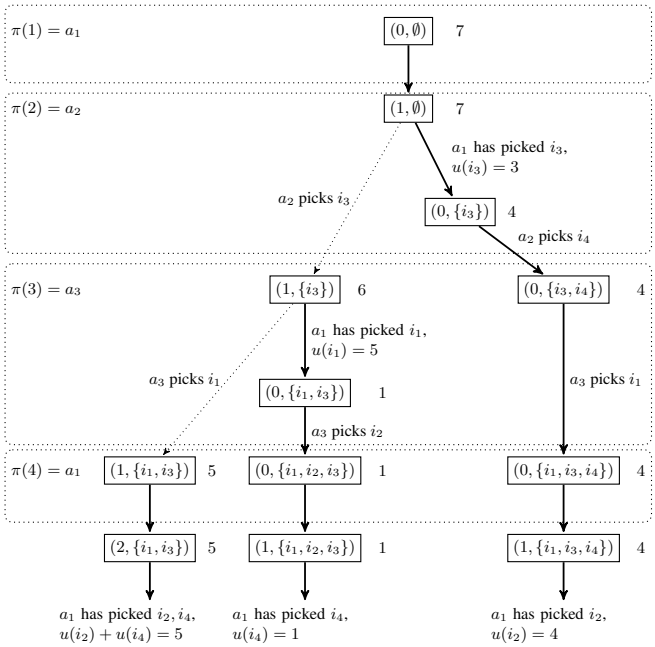

Figure 1. Directed acyclic state graph $G_{\mathtt{dp}}$ in Example 1

unchanged. Otherwise, let $\pi(|S| + k + 1) = a^*$, then the successor states are $(k, S \cup \{b(a^*, S)\})$ and $(k-1, S \cup \{b(a^*, S)\})$. Then we have the following two inclusion relationships:

$$D(a^*, b(a^*, S)) \cup \{b(a^*, S)\} \subseteq D(a^*, b(a^*, S \cup \{b(a^*, S)\})),$$
$$\forall a \in \mathcal{A} \setminus \{a_1\}, D(a, b(a, S)) \subseteq D(a, b(a, S \cup \{b(a^*, S)\})).$$

These relationships imply that $S \cup \{b(a^*, S)\}$ is equal to:

$$\bigcup_{a \in \mathcal{A} \setminus \{a_1\}} D(a, b(a, S)) \cup \{b(a^*, S)\} \subseteq \bigcup_{a \in \mathcal{A} \setminus \{a_1\}} D(a, b(a, S \cup \{b(a^*, S)\})),$$

by the induction hypothesis. As already stated, the reverse inclusion relationship is always true and hence $S \cup \{b(a^*, S)\} \in \Delta$. □

Consequently from Lemma 1, each state in $V_{\mathtt{dp}}$ admits two possible representations that we call *agent representation* and *item representation*. In the item representation, a state (k, S) is represented by a vector of size $m + 1$, i.e., S is represented by a binary vector of size m. In the agent representation, a state (k, S) is represented by a vector of size n. In this case, S is replaced by the vector $(b(a_2, S), \ldots, b(a_n, S))$.[2] Note that processing a state (computing the successor states and the optimal value of the state according to the ones of the successor states) in the agent (resp. item) representation can be done in $O(nm)$ (resp. $O(m)$) operations. We now show that the $ManipSA$ problem can be solved in polynomial time for any bounded number of agents.

Theorem 1. *Problem $ManipSA$ is solvable in $O(n \cdot m^{n+1})$. As a result, $ManipSA$ is in XP w.r.t. parameter n.*

Proof. The result follows from the fact that our dynamic programming method runs in $O(n \cdot m^{n+1})$. To obtain this complexity bound, one should use the agent representation. In this case, processing a state requires $O(nm)$ operations, and one can use a dynamic programming table of size m^n with one cell per possible vector $(k, b(a_2, S), \ldots, b(a_n, S))$. □

We now argue that our dynamic programming approach yields an FPT algorithm w.r.t. parameters $n + \mu(a_1)$, $n + \mathtt{rg}^{\max}$ and $\mathtt{rg}^{\max}$ by providing tighter upper bounds on $|\mathcal{S}_{\mathtt{dp}}|$. To use these bounds, we will need the two following lemmata:

Lemma 2. *Under the item representation, the graph $G_{\mathtt{dp}}$ can be build in $O(m|V_{\mathtt{dp}}|^2)$.*

Proof. First note that $G_{\mathtt{dp}}$ is indeed acyclic. Indeed, given a state that can occur at time step t of the allocation process (i.e., $k + |S| = t$), its successors will either correspond to time step $t + 1$ or will still correspond to time step t but with a strictly lower value for parameter k. We now show how to incrementally build $G_{\mathtt{dp}}$ from state $(k = 0, S = \emptyset)$. For each new state generated at the previous iteration, compute its successor states, add edges towards them, and label them with the corresponding utility values. Moreover, each time a state is generated, compare it to the states already generated to avoid the creation of duplicates. If it is indeed a new state, its successors will be computed in the next iteration. This process is repeated until all states are generated. Note that because each state is only processed once, we will generate at most $2|V_{\mathtt{dp}}|$ states. However, because of the duplicate removal operation performed each time a state is generated, the method runs in $O(m|V_{\mathtt{dp}}|^2)$. Indeed, this step will trigger $O(|V_{\mathtt{dp}}|^2)$ comparisons, each requiring $m + 1$ operations. □

Lemma 3. *Under the item representation, problem $ManipSA$ can be solved in $O(m|V_{\mathtt{dp}}|^2)$.*

Proof. By Lemma 2, we can build $G_{\mathtt{dp}}$ in $O(m|V_{\mathtt{dp}}|^2)$ and compute an optimal manipulation by backward induction in $O(m|V_{\mathtt{dp}}|)$. □

The algorithm is FPT w.r.t. parameter $n + \mu(a_1)$. We further argue that $|\mathcal{S}_{\mathtt{dp}}|$ can be upper bounded by $m(\mu(a_1) + 1)^{n-1}$. This is a consequence of the following lemma, where $\mu(a, t)$ denotes the number of items that agent a gets to pick within the t first time steps.

Lemma 4. *For each time step t, there is a set S_t of $t - \mu(a_1, t)$ items that are always picked within the t first time steps, whatever the actions of the manipulator.*

Sketch of the proof. Given an instance $\mathcal{J}$ of the $ManipSA$ problem, consider the instance $\mathcal{J}^{-a_1}$ obtained from $\mathcal{J}$ by removing a_1. Moreover, let us denote by $S_t^{-a_1}$ the set of items picked at the end of the t^{th} time step in $\mathcal{J}^{-a_1}$. This set of size t is clearly defined as all agents behave truthfully in $\mathcal{J}^{-a_1}$. We argue that after t time steps in $\mathcal{J}$, all items in $S^{-a_1}_{t-\mu(a_1,t)}$ have been picked whatever the actions of a_1. This can be showed by induction because at each time step where the picker is a non-manipulator, she will pick the same item as in $\mathcal{J}^{-a_1}$ unless this item has already been picked. □

As a consequence of Lemma 4, for each possible set S that can appear at time step t and agent $a \in \mathcal{A} \setminus \{a_1\}$, $b(a, S)$ can only be $\mu(a_1, t-1)+1$ different items. More precisely, $b(a, S)$ has to be one of the $\mu(a_1, t-1)+1$ preferred items of a in $\mathcal{I} \setminus S_{t-1}$. As a result, the number of possible vectors $(b(a_2, S), \ldots, b(a_n, S))$ associated to all the possible sets S that can appear at time step t is upper bounded by $(\mu(a_1, t-1) + 1)^{n-1}$. Lastly, by using the facts that each set $S \in \mathcal{S}_{\mathtt{dp}}$ is characterized by the vector $(b(a_2, S), \ldots, b(a_n, S))$, that $\mu(a_1, t) \leq \mu(a_1)$ for all t, and by considering all possible time steps, we obtain that $|\mathcal{S}_{\mathtt{dp}}| \leq m(\mu(a_1) + 1)^{n-1}$.

[2] When proving bounds on $|\mathcal{S}_{\mathtt{dp}}|$, we will consider that values $b(a_i, S)$ belong to $\mathcal{I}$, not considering the one state where all items are already picked.

Theorem 2. *Problem ManipSA is solvable in $O(m^3(\mu(a_1)+1)^{2n})$. As a result, ManipSA is FPT w.r.t. parameter $n+\mu(a_1)$.*

Proof. This result is a consequence of Lemma 3 and the fact that $|V_{\mathtt{dp}}| \leq (\mu(a_1)+1)|\mathcal{S}_{\mathtt{dp}}| \leq m(\mu(a_1)+1)^n$. □

The algorithm is FPT w.r.t. parameter $n+\mathtt{rg}^{\max}$. We show that $|\mathcal{S}_{\mathtt{dp}}|$ is also upper bounded by $m(2\mathtt{rg}^{\max})^{n-2}$. This is a consequence of the following lemma.

Lemma 5. *For any set $S \subseteq \mathcal{I}$, and any two agents $a_s, a_t \in \mathcal{A}\backslash\{a_1\}$,*

$$|\mathtt{rk}_{a_s}(b(a_s,S)) - \mathtt{rk}_{a_t}(b(a_t,S))| \leq \mathtt{rg}^{\max} - 1.$$

Proof. If we assume for the sake of contradiction that $\mathtt{rk}_{a_s}(b(a_s,S)) \geq \mathtt{rk}_{a_t}(b(a_t,S)) + \mathtt{rg}^{\max}$ and use the fact that $|\mathtt{rk}_{a_s}(b(a_t,S)) - \mathtt{rk}_{a_t}(b(a_t,S))| < \mathtt{rg}^{\max}$ (by definition of $\mathtt{rg}^{\max}$), then we can conclude that $b(a_t,S) \succ_{a_s} b(a_s,S)$, which contradicts the definition of $b(a_s,S)$. □

Lemma 5 implies that for each of the m possible items for $b(a_2,S)$, there are only $2\mathtt{rg}^{\max}-1$ possible items for other parameters $b(a_j,S)$ with $j>2$. Then, by using the facts that a set $S \in \mathcal{S}_{\mathtt{dp}}$ is characterized by the vector $(b(a_2,S),\ldots,b(a_n,S))$, we obtain that $|\mathcal{S}_{\mathtt{dp}}| \leq m(2\mathtt{rg}^{\max})^{n-2}$.

Theorem 3. *Problem ManipSA is solvable in $O(m^5(2\mathtt{rg}^{\max})^{2(n-2)})$. As a result, ManipSA is FPT w.r.t. parameter $n+\mathtt{rg}^{\max}$.*

Proof. This result is a consequence of Lemma 3 and the fact that $|V_{\mathtt{dp}}| \leq m|\mathcal{S}_{\mathtt{dp}}| \leq m^2(2\mathtt{rg}^{\max})^{n-2}$. □

The algorithm is FPT w.r.t. parameter $\mathtt{rg}^{\max}$. Lastly, $|\mathcal{S}_{\mathtt{dp}}|$ can also be upper bounded by $m2^{2\mathtt{rg}^{\max}}$. This claim is due to the fact that the set $S \setminus D(a_2,b(a_2,S))$ cannot contain an item whose rank w.r.t. a_2 is "too high", which is proved in the following lemma.

Lemma 6. *Given $S \in \mathcal{S}_{\mathtt{dp}}$, all i in $S \setminus D(a_2,b(a_2,S))$ verify*

$$\mathtt{rk}_{a_2}(b(a_2,S)) + 1 \leq \mathtt{rk}_{a_2}(i) \leq \mathtt{rk}_{a_2}(b(a_2,S)) + 2\mathtt{rg}^{\max}.$$

Proof. First note that by definition $b(a_2,S) \notin S$ (because $b(a_2,S) \in \mathcal{I}\setminus S$) and $D(a_2,b(a_2,S)) = \{i \in \mathcal{I} | \mathtt{rk}_{a_2}(i) < \mathtt{rk}_{a_2}(b(a_2,S))\}$. Hence, the first inequality of the lemma holds.

Let us assume for the sake of contradiction that there exists $i \in S \setminus D(a_2,b(a_2,S))$ such that $\mathtt{rk}_{a_2}(i) > \mathtt{rk}_{a_2}(b(a_2,S)) + 2\mathtt{rg}^{\max}$. Because S belongs to Δ, we have that $S \setminus D(a_2,b(a_2,S)) = \bigcup_{a\in\mathcal{A}\setminus\{a_1\}} D(a,b(a,S)) \setminus D(a_2,b(a_2,S))$. Hence, there exists a_j with $j \geq 3$ such that $i \in D(a_j,b(a_j,S))$. By definition of $\mathtt{rg}^{\max}$, we have that $\mathtt{rg}^{\max} > \mathtt{rk}_{a_2}(i) - \mathtt{rk}_{a_j}(i)$, or equivalently that $\mathtt{rk}_{a_j}(i) > \mathtt{rk}_{a_2}(i) - \mathtt{rg}^{\max}$, which yields that $\mathtt{rk}_{a_j}(b(a_j,S)) > \mathtt{rk}_{a_j}(i) > \mathtt{rk}_{a_2}(i) - \mathtt{rg}^{\max} > \mathtt{rk}_{a_2}(b(a_2,S)) + \mathtt{rg}^{\max}$. This contradicts Lemma 5. □

As a consequence of Lemma 6, $|\mathcal{S}_{\mathtt{dp}}|$ is upper bounded by $m2^{2\mathtt{rg}^{\max}}$ because there are at most m possible items for $b(a_2,S)$, and for each of them, there are at most $2^{2\mathtt{rg}^{\max}}$ possible sets for $S \setminus D(a_2,b(a_2,S))$.

Theorem 4. *Problem ManipSA is solvable in $O(m^5 2^{4\mathtt{rg}^{\max}})$. As a result, ManipSA is FPT w.r.t. parameter $\mathtt{rg}^{\max}$.*

Proof. This result is a consequence of Lemma 3 and the fact that $|V_{\mathtt{dp}}| \leq m|\mathcal{S}_{\mathtt{dp}}| \leq m^2 2^{2\mathtt{rg}^{\max}}$. □

Remark. Note that it is easy to prove that, in contrast, the problem is NP-hard even if the average range of the items is less than or equal to 2.[3] Furthermore, Theorem 3 might seem less appealing as the *ManipSA* problem is FPT w.r.t. parameter $\mathtt{rg}^{\max}$ alone. However, we would like to stress that the time complexity of Theorem 3 might be more interesting than the one of Theorem 4 for a small number of agents.

4 PARAMETERIZED HARDNESS RESULTS

We have seen in the last section that the *ManipSA* problem is in XP w.r.t. parameters n and $\mu(a_1)$ and that it is in FPT for parameters $\mathtt{rg}^{\max}$ and $n+\mu(a_1)$. One could hope for more positive results for parameters n and $\mu(a_1)$, as an FPT algorithm. However, we show in this section that the *ManipSA* problem is W[1]-hard w.r.t. each of these two parameters.

We start with the W[1]-hardness result on parameter $\mu(a_1)$. In fact, we obtain a stronger result by proving that even determining if there exists a successful manipulation is W[1]-hard w.r.t. $\mu^{\max}$.

Theorem 5. *Determining if there exists a successful manipulation for a_1 is W[1]-hard w.r.t. parameter $\mu^{\max}$.*

Proof. We sketch here the main ideas of the proof. We make a parameterized reduction from the CLIQUE problem where given a graph $G=(V,E)$ and an integer k, we wish to determine if there exists a clique of size k. This problem is W[1]-hard w.r.t. parameter k. W.l.o.g., we make the assumptions that $|V|>k$ and that $|E|>k(k-1)/2$ (because otherwise it is trivial to determine if there is a clique of size k).

From an instance of CLIQUE $(G=(V,E),k)$, we create the following *ManipSA* instance.

Set of items. We create two items, $g_{\{i,j\}}$ (a good item) and $w_{\{i,j\}}$ (one of the worst items), for each edge $\{i,j\} \in E$ and two items, b_i (one of the best items) and m_i (a medium item), for each vertex $i \in V$. Put another way, $\mathcal{I} = \{g_{\{i,j\}}, w_{\{i,j\}} | \{i,j\} \in E\} \cup \{b_i, m_i | i \in V\}$ and the number of items is thus $|\mathcal{I}| = 2|V| + 2|E|$.

Set of agents. We create one agent $e_{\{i,j\}}$ for each edge $\{i,j\} \in E$ and one agent v_i for each vertex $i \in V$. The top of $e_{\{i,j\}}$'s ranking is $g_{\{i,j\}} \succ m_i \succ m_j \succ w_{\{i,j\}}$ (which one of m_i or m_j is ranked first can be chosen arbitrarily). The top of v_i's ranking is $b_i \succ m_i$. We also create $|V|-k-1$ agents c_t for $t \in \{1,\ldots,|V|-k-1\}$ (whose role is to collect medium items) such that the top of the ranking of each c_t is $m_1 \succ m_2 \ldots \succ m_{|V|}$. Last, the manipulator, that we denote by a_1 to be consistent with the rest of the paper, has the following preferences: he first ranks items b_i, then items $g_{\{i,j\}}$, then items m_i and last items $w_{\{i,j\}}$. To summarize, $\mathcal{A} = \{e_{\{i,j\}} | \{i,j\} \in E\} \cup \{v_i | i \in V\} \cup \{c_t | t \in \{1,\ldots,|V|-k-1\}\} \cup \{a_1\}$ and there are $|\mathcal{A}| = 2|V| + |E| - k$ agents.

Picking sequence. The picking sequence π is composed of the following rounds:

- Manipulator round 1: a_1 gets to pick k items.
- Vertex round: each agent v_i gets to pick one item.

[3] One can use a reduction with a sufficiently large number of dummy items ranked last and in the same positions by all agents.

- Manipulator round 2: a_1 gets to pick $k(k-1)/2$ items.
- Edge round: each agent $e_{\{i,j\}}$ gets to pick one item.
- Medium item collectors round: each agent c_t gets to pick one item.
- Manipulator round 3: a_1 gets to pick one item.
- End round: the remaining items are shared arbitrarily within the non-manipulators such that each agent gets at most one new item.

Note that $\mu^{\max} = \mu(a_1) = \frac{k(k+1)}{2} + 1$.

Utility values of a_1. For ease of presentation, we will act as if there were only four different utility values, even if a_1 is asked to report a complete preference order. This simplification can be removed by using a similar utility function with sufficiently small epsilon values to make preferences strict. In this sketch of proof, each item b_i has utility 4. Each item $g_{\{i,j\}}$ has utility 3. Each item m_i has utility 2. Lastly, items $w_{\{i,j\}}$ have utility 1. In this simplified setting, we set $\succ_T$ as being one specific ranking consistent with the utility values of a_1 and we wish to determine if there exists another ranking yielding a strictly higher utility.

Sketch of the proof. At the end of the vertex round, all the best items are gone, as they have already been picked by a_1 or by the vertex agents v_i. Similarly, at the end of the edge round, none of the good items are left. Hence, at the third manipulator round, when a_1 picks her last item, the best she can hope for is a medium item. Consequently, the maximum utility she might achieve is accomplished by picking k best items in her first round, $k(k-1)/2$ good items in her second round, and finally a medium item in her third round, for an overall utility of $4k+3k(k-1)/2+2$. Note that she can always pick any set of k best items in her first round and then (whatever the previous k best items) pick any set of $k(k-1)/2$ good items in her second round. Hence obtaining an overall utility of $4k+3k(k-1)/2+1$ is always possible. Note also that, if $\{b_{i_1}, \ldots, b_{i_k}\}$ are the k best items selected by a_1 at the first round, then in the vertex round the vertex agents $\{v_{i_1}, \ldots, v_{i_k}\}$ will pick the medium items $\{m_{i_1}, \ldots, m_{i_k}\}$. Moreover, before the third manipulator round, agents c_t will pick additional $|V| - k - 1$ medium items. So, a medium item is left at the third manipulator round only if none of the edge agents picks a medium item in the edge round. According to her preference ranking, any such agent $e_{\{i,j\}}$ will not pick a medium item iff $g_{\{i,j\}}$ is still available, or if $g_{\{i,j\}}$, m_i and m_j have all already been picked. If $g_{\{i,j\}}$ is one of the $k(k-1)/2$ good items that have been already picked at the manipulator second round, then m_i and m_j have already been picked before by v_i and v_j in the vertex round, if b_i and b_j were already taken in the first manipulator round. In conclusion, none of the medium items are picked by the edge agents iff the $k(k-1)/2$ edges $e_{\{i,j\}}$ for which $g_{\{i,j\}}$ has already been picked at the second manipulator round have as endpoints only nodes in $\{v_{i_1}, \ldots, v_{i_k}\}$, and this is possible iff $\{v_{i_1}, \ldots, v_{i_k}\}$ forms a clique in the initial graph G. Summarizing, there exists a strategy for a_1 achieving an overall utility of $4k+3k(k-1)/2+2$ iff G has a clique of k nodes. It remains to show that we could solve the CLIQUE problem if we could determine if there exists a successful manipulation. This fact results from the following disjunction of two cases: If $u_T = 4k+3k(k-1)/2+2$, then we can conclude that there exists a clique of size k; Otherwise, $u_T = 4k+3k(k-1)/2+1$ and there exists a clique of size k iff there exists a successful manipulation. □

Remark: Aziz et al. [1] considered a sequential allocation setting in which the manipulator has a binary utility function, but is asked to provide a complete preference order. In this setting, the manipulation problem consists in finding a ranking maximizing the utility of the bundle she gets. While the authors showed that this problem (with binary utilities) can be solved in polynomial time, the reduction used in the proof of Theorem 5 shows that it is NP-hard if the manipulator has a utility function involving four different values (instead of two).

Similarly, we obtain that the $ManipSA$ problem is W[1]-hard w.r.t. the number of agents.

Theorem 6. *$ManipSA$ is W[1]-hard w.r.t. the number of agents.*

Proof. We design a parameterized reduction from MULTICOLORED CLIQUE. In this problem, given a graph $G = (V, E)$ with vertex set $V = \{v_1, \ldots, v_n\}$, an integer k, and a vertex coloring $\varphi : V \to \{1, \ldots, k\}$, we wish to determine if there exists a clique of size k in G containing exactly one vertex per color. MULTICOLORED CLIQUE is known to be W[1]-hard w.r.t. parameter k [13].

Idea of the proof: We resort on the nice mathematical tool of Sidon sequences. These sequences associate to each number i in $\{1, \ldots, n\}$ a value $id(i)$ such that, for every pair (i, l) with $i \leq l$, the sum $id(i) + id(l)$ is different from the one of any other different pair of elements in $\{1, \ldots, n\}$. We use the construction of Erdös and Turàn [12], by setting $id(i) = 2 \cdot p \cdot i + (i^2 \mod p)$ for every $i \in \{1, \ldots, n\}$, where p is the smallest prime number greater than n. Notice that, by the Bertrand-Chebyshev theorem [11], $p < 2n$, and thus $id(i) = O(n^2)$. This sequence will be used in the following way. We create a large set of items B_j for each color j. In the first picking round, the manipulator will be able to pick a large number of items within these sets. To recover a solution of the MULTICOLORED CLIQUE problem, we will show that, if a multicolored clique $\{v_{i_1}, \ldots, v_{i_k}\}$ exists in which each vertex v_{i_j} has color $\varphi(v_{i_j}) = j$, then in an optimal manipulation the manipulator should pick exactly $(k+1) \cdot id(i_j)$ items in each set B_j. The edges of the clique will then be identified by the sums $id(i_j) + id(i_r)$ for all pairs of vertices $\{v_{i_j}, v_{i_r}\} \subset \{v_{i_1}, \ldots, v_{i_k}\}$.

From a MULTICOLORED CLIQUE instance $(G = (V, E), k, \varphi)$, we construct the following $ManipSA$ instance.

Set of items:

- For each color j, we create a set B_j of $(k+1) \cdot id(n)$ items, and two sets Id_j and $Id_{\overline{j}}$ of $id(n) + 2$ items each. The purpose of items in $Id_j \cup Id_{\overline{j}}$ is to ensure that the number of items picked by a_1 in B_j is of the form $(k+1) \cdot id(i)$ such that $\varphi(v_i) = j$.
- For each pair of colors $\{j, r\}$ with $j \neq r$, we create a set $Id_{\{j,r\}}$ of $2 \cdot (id(n) + 1)$ items. The purpose of items in $Id_{\{j,r\}}$ is to ensure that, whenever a_1 picks $(k+1) \cdot id(i)$ items in B_j and $(k+1) \cdot id(l)$ items in B_r for two given vertices v_i and v_l of colors $\varphi(v_i) = j$ and $\varphi(v_l) = r$, then $\{v_i, v_l\}$ is an edge of G.
- We create a set D of $k(k+1) \cdot id(n)$ items. In a first picking round, the manipulator will be able to pick items in $\bigcup_{j=1}^{k} B_j \cup D$. The purpose of items in D is to make it possible for a_1 to adjust the number of items she picks in $\bigcup_{j=1}^{k} B_j$.
- Last, we add a set Z of $2k(k+1)id(n)$ items. Set Z will be used as a buffer of items where each non-manipulator will pick when no items in Id_j, $Id_{\overline{j}}$, or $Id_{j,r}$ are left, so as to avoid mutual conflicts.

Set of agents: We create two agents c_j and $\overline{c}_j$ per color j, and two agents $p_{j,r}$ and $p_{r,j}$ for each pair of colors $\{j, r\}$ such that $j \neq r$. Moreover, we create one agent denoted by d and one manipulator a_1. In total, there are $k(k+1) + 2$ agents. We now detail the top of the preference rankings of non-manipulators, where by abuse of

notations, we use $S \succ S'$ to denote the fact that items in S are ranked before the ones in S', while the order inside each set is indifferent.

- Agent c_j, for $1 \leq j \leq k$: $B_j \succ Id_j \succ Z \succ \ldots$.
- Agent $\overline{c}_j$, for $1 \leq j \leq k$: $B_j \succ Id_{\overline{j}} \succ Z \succ \ldots$.
- Agent $p_{j,r}$, for $1 \leq j \neq r \leq k$: $B_j \succ Id_{\{j,r\}} \succ Z \succ \ldots$.
- Agent $p_{r,j}$ for $1 \leq j \neq r \leq k$: $B_r \succ Id_{\{j,r\}} \succ Z \succ \ldots$.
- Agent d: $D \succ Z \succ \ldots$.

As an important remark, note that agents $p_{j,r}$ and $p_{r,j}$ rank items in $Id_{\{j,r\}}$ identically.

Picking sequence: π is composed of the following rounds:

- Manipulator round 1: a_1 gets to pick $k(k+1) \cdot id(n)$ items.
- Non-manipulators round:
  - Agents in $\mathcal{A} \setminus \{a_1, d\}$ pick in $id(n)$ subrounds. In each subround, each of them picks exactly one item in the following order: agents c_j, $1 \leq j \leq k$, are the first pickers, then come agents $p_{j,r}$, and lastly agents $\overline{c}_j$.
  - Finally, agent d picks $k(k+1) \cdot id(n)$ items.
- Manipulator round 2: a_1 gets all remaining items.

Utility values of a_1: For ease of presentation, we use two simplifying assumptions. First, we act as if different items can have the same utility value for a_1. This assumption can be removed making preferences strict by adding sufficiently small epsilon values. Second, we use negative utilities. In fact, one can recover an equivalent instance with only non-negative values by adding to all the utilities the absolute value of the minimal one. Indeed, this would not change the set of optimal solutions as the size of a_1's bundle is fixed by π.

- Items in Z have a utility value of 0.
- One specific item in each set B_j, that we denote by b_j^*, has a utility value of 4α where $\alpha = (id(n)+2)k(k+1)$. All items in $(\bigcup_{j=1}^{k} B_j \cup D) \setminus \{b_1^*, \ldots, b_k^*\}$ have a utility value equal to 2α.
- The utilities of the items in the sets Id_j and $Id_{\overline{j}}$ are defined as follows. Index the items in Id_j (resp. $Id_{\overline{j}}$) from 1 to $id(n)+2$ according to the preference order of agent c_j (resp. $\overline{c}_j$). Furthermore, let $\mathcal{T}_j = \{id(i) | \varphi(v_i) = j\}$, $\tau_j(t)$ denote the t^{th} smallest value in $\mathcal{T}_j$, and $T_j = |\mathcal{T}_j|$. We also set $\tau_j(0) = 0$ and $\tau_j(T_j+1) = id(n)+2$. Then, all items receive a utility value of 1, except for the items of indices $\tau_j(t)$ for $t \in \{1, \ldots, T_j+1\}$, that get utility $\tau_j(t-1) - \tau_j(t) + 1$. Notice that, for every t such that $1 \leq t \leq T_j+1$, by definition the sum of the utilities of all the items from $\tau_j(t-1)+1$ to $\tau_j(t)$ is 0.
- Similarly, the utilities of the items in each $Id_{\{j,r\}}$ are set in the following manner. Index these items from 1 to $2id(n)+2$ according to the preference order of agents $p_{j,r}$ and $p_{r,j}$. Furthermore, let $\mathcal{T}_{j,r} = \{id(i) + id(l) | \varphi(v_i) = j, \varphi(v_l) = r, \{v_i, v_l\} \in E\}$, $\tau_{j,r}(t)$ denote the t^{th} smallest value in $\mathcal{T}_{j,r}$, and $T_{j,r} = |\mathcal{T}_{j,r}|$. We also set $\tau_{j,r}(0) = 0$ and $\tau_{j,r}(T_{j,r}+1) = 2id(n)+2$. Then all items receive a utility value of 1, except items of index $\tau_{j,r}(t)$ for $t \in \{1, \ldots, T_{j,r}+1\}$, whose utility is set to $\tau_{j,r}(t-1) - \tau_{j,r}(t) + 1$.

As we are going to show below, in an optimal manipulation, the agents behave as follows. In the first manipulator round, a_1 picks $k(k+1) \cdot id(n)$ items in $\bigcup_j B_j \cup D$. Then, in the non-manipulators round, agents c_j, $p_{j,r}$ and $\overline{c}_j$ for the different values j and $r \neq j$ pick the remaining items in the sets B_j, plus other items in Id_j, $Id_{\overline{j}}$ and $Id_{\{j,r\}}$. Subsequently, d takes all the remaining items in D and further ones to complete her picks in Z. Finally, in the second manipulator round, a_1 collects all remaining items.

Sketch of the proof. We first claim that, in the first manipulator round, a_1 should pick only items in $\bigcup_j B_j \cup D$. Indeed, after the non-manipulators round, none of these items is left, whatever a_1 has previously picked. In particular, the items left by a_1 in each set B_j are collected by agents c_j, $p_{j,r}$ and $\overline{c}_j$, while the ones in D are collected by agent d. Moreover, because $|\bigcup_j (Id_j \cup Id_{\overline{j}}) \cup \bigcup_{j \neq r} Id_{j,r}|$ is upper bounded by α and of the utility function we have set, any subset of items in $\mathcal{I} \setminus (\bigcup_j B_j \cup D)$ as a utility value which is strictly less than α and strictly greater than $-\alpha$. As a result, because each item in $\bigcup_j B_j \cup D$ is worth 2α, any solution which would not pick only items in $\bigcup_j B_j \cup D$ in the first manipulator round could be improved by doing so. Using the same type of argument, we also claim that a_1 should pick all of the b_j^* items in her first picking round. We will hence restrict our attention to picking strategies that verify these two assumptions. Under such an hypothesis, the best utility value a_1 can hope to get from the set of items she collects in her second picking round is 0. This is induced by the utility values that we have set, as well as by the truthful picking strategies of non-manipulators. Indeed, note that by construction the overall utility of the set $\mathcal{I} \setminus (\bigcup_j B_j \cup D)$ is 0. Moreover, as sets Id_j, $Id_{\overline{j}}$ and $Id_{\{j,r\}}$ are indexed according to the preference orders of agents c_j, $\overline{c}_j$, $p_{j,r}$ and $p_{r,j}$, at the end of the non-manipulators round only prefixes of such sets have been picked. Hence, recalling that all the items in Z have null utility for a_1, the overall utility of items left to a_1 at the beginning of the second manipulator round is 0 if and only if the prefixes of the already picked items in all the sets Id_j, $Id_{\overline{j}}$ and $Id_{\{j,r\}}$ end up to items of negative value for a_1. We now argue that this can happen if and only if there exists a multicolored clique G.

Let us first show the only if direction, i.e., that if a_1 gets an overall utility equal to 0 from the set of items she collects in her second picking round, then there is a multicolored clique of size k in G. Let us denote by nb_j the number of items that a_1 has picked in B_j during the first manipulator round. Since for each $j \in \{1, \ldots, k\}$ agent a_1 has picked b_j^* and $|B_j| = (k+1)id(n)$, we have that $1 \leq nb_j \leq (k+1)id(n)$. We first show that nb_j should be a multiple of $k+1$.

To this aim, let us first observe that, for each $j \in \{1, \ldots, k\}$, after the first manipulator round, in every non-manipulators subround, $k+1$ items of B_j (if still available) are picked by the $k+1$ agents c_j, $p_{j,r}$ with $j \neq r$ and $\overline{c}_j$ (in this order). Therefore, at the end of the non-manipulators rounds, c_j has picked $\lfloor nb_j/(k+1) \rfloor$ items in Id_j and $\overline{c}_j$ has picked $\lceil nb_j/(k+1) \rceil$ items in $Id_{\overline{j}}$. But then, if nb_j is not a multiple of $k+1$, these two numbers are different and thus the last items picked by c_j in Id_j and by $\overline{c}_j$ in $Id_{\overline{j}}$ cannot both have negative utility for a_1, because the difference between two consecutive id values is strictly greater than 1. Therefore, each nb_j should be of the form $nb_j = (k+1) \cdot id(i_j)$ for some $i_j \in \{1, \ldots, n\}$ such that $\varphi(v_{i_j}) = j$, so that both c_j and $\overline{c}_j$ pick $id(i_j)$ items in Id_j and $Id_{\overline{j}}$, respectively. In order to show that $\{v_{i_j} | 1 \leq j \leq k\}$ is a multicolored clique, it remains to prove that all the vertices of this set are neighbors in G. Indeed, since in each subround of the non-manipulators round every time c_j picks in $Id(j)$ each agent $p_{j,r}$ picks in $Id_{\{j,r\}}$, at the end of the non-manipulators round $p_{j,r}$ and $p_{r,j}$ have picked $id(i_j) + id(i_r)$ items in $Id_{\{j,r\}}$. Since in order for a_1 to achieve an overall utility equal to 0 in the second manipulator round the last item previously picked in $Id_{\{j,r\}}$ must have a negative utility, $\{v_{i_j}, v_{i_r}\}$ must be an edge of G. To summarize, the correspondence between an edge $\{v_i, v_j\}$ and the preferences of the agents in the resulting $ManipSA$ instance is illustrated in Figure 2,

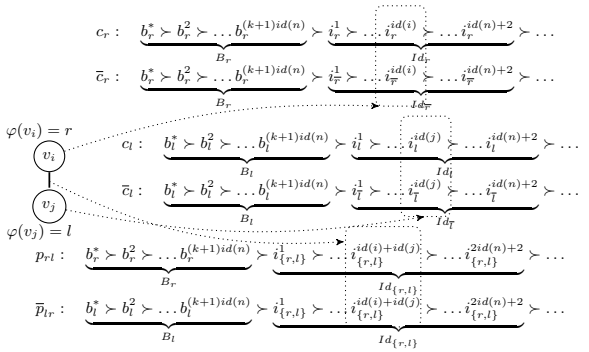

Figure 2. Correspondence between an edge in MULTICOLORED CLIQUE and agents' preferences in the resulting $ManipSA$ instance.

where we have circled the last items picked by the agents in a solution of the $ManipSA$ problem corresponding to a solution of the MULTICOLORED CLIQUE instance containing both v_i and v_j.

It remains to show the if direction, i.e., that if there is a multicolored clique $\{v_{i_1}, \ldots, v_{i_k}\}$ in G, then there exists a strategy leading a_1 to reach overall utility 0 in her second manipulation round. Assuming without loss of generality that $\varphi(v_{i_j}) = j$, this can be accomplished by letting a_1 pick $nb_j = (k+1) \cdot id(i_j)$ items in B_j, $1 \leq j \leq k$, and the remaining items in D. Then, each c_j (resp. $\overline{c}_j$) will pick $id(i_j)$ items in $Id(j)$ (resp. $Id_{\overline{j}}$) and each $p_{j,r}$ will pick $id(i_j)$ items in $Id_{\{j,r\}}$, which causes a_1 to achieve overall utility 0 in her second manipulator round, finally proving the claim. □

Consequently from Theorems 5 and 6, it is unlikely that the $ManipSA$ problem admits FPT algorithms w.r.t. parameters $\mu(a_1)$ and n. Hence, these results valorize the XP results on these parameters obtained in Section 3, as well as Theorem 2, which interestingly shows that the $ManipSA$ problem is FPT w.r.t. parameter $\mu(a_1)+n$.

5 CONCLUSION AND FUTURE WORK

We have provided a complete characterization of the parameterized complexity of finding an optimal manipulation in the sequential allocation protocol w.r.t. some fundamental parameters and their combinations. In particular, we have given FPT and XP results, showing the tightness of the XP ones by means of corresponding W[1]-hardness proofs. Some of our positive results are obtained through a novel dynamic programming algorithm, which might be potentially useful when considering other parameters.

Several directions for future work are conceivable, like investigating other parameters and estimating the price of manipulation, i.e., a worst case ratio measuring the loss in social welfare caused by one agent manipulating, all the others being truthful. Moreover, it would be interesting to investigate the sequential allocation protocol under different realistic assumptions concerning the utility functions and an incomplete information by the manipulator.

ACKNOWLEDGEMENTS

This work has been partially supported by the Italian MIUR PRIN 2017 Project ALGADIMAR "Algorithms, Games, and Digital Markets". We are grateful to Jérôme Lang and Paolo Serafino for helpful comments and stimulating conversations on this work.

REFERENCES

[1] Haris Aziz, Sylvain Bouveret, Jérôme Lang, and Simon Mackenzie, 'Complexity of manipulating sequential allocation', in *Proceedings of the 31st AAAI Conference on Artificial Intelligence, AAAI 2017, February 4-9, 2017, San Francisco, California, USA.*, pp. 328–334, (2017).

[2] Haris Aziz, Paul Goldberg, and Toby Walsh, 'Equilibria in sequential allocation', in *Proceedings of the 5th International Conference on Algorithmic Decision Theory, ADT 2017, October 25-27, 2017, Luxembourg, Luxembourg*, pp. 270–283, (2017).

[3] Haris Aziz, Thomas Kalinowski, Toby Walsh, and Lirong Xia, 'Welfare of sequential allocation mechanisms for indivisible goods', in *Proceedings of the 22nd European Conference on Artificial Intelligence, ECAI 2016, 29 August-2 September, 2016, The Hague, The Netherlands*, pp. 787–794, (2016).

[4] Haris Aziz, Toby Walsh, and Lirong Xia, 'Possible and necessary allocations via sequential mechanisms', in *Proceedings of the 24th International Joint Conference on Artificial Intelligence, IJCAI 2015, July 25-31, 2015, Buenos Aires, Argentina*, pp. 468–474, (2015).

[5] Sylvain Bouveret, Yann Chevaleyre, and Nicolas Maudet, 'Fair allocation of indivisible goods', in *Handbook of Computational Social Choice*, 284–310, Cambridge University Press, (2016).

[6] Sylvain Bouveret and Jérôme Lang, 'A general elicitation-free protocol for allocating indivisible goods', in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence, IJCAI 2011, July 16-22, 2011, Barcelona, Catalonia, Spain*, pp. 73–78, (2011).

[7] Sylvain Bouveret and Jérôme Lang, 'Manipulating picking sequences', in *Proceedings of the 21st European Conference on Artificial Intelligence, ECAI 2014, 18-22 August, 2014, Prague, Czech Republic*, pp. 141–146, (2014).

[8] Steven J Brams, Paul H Edelman, and Peter C Fishburn, 'Fair division of indivisible items', *Theory and Decision*, **55**(2), 147–180, (2003).

[9] Steven J Brams and Philip D Straffin Jr, 'Prisoners' dilemma and professional sports drafts', *The American Mathematical Monthly*, **86**(2), 80–88, (1979).

[10] Eric Budish and Estelle Cantillon, 'Strategic behavior in multi-unit assignment problems: Theory and evidence from course allocations', in *Computational Social Systems and the Internet, 1.7. - 6.7*, (2007).

[11] Pafnuty Lvovich Chebyshev, 'Mémoire sur les nombres premiers', *Journal de Mathématiques Pures et Appliquées*, **17**, 366–390, (1852).

[12] Paul Erdös and Pál Turán, 'On a problem of sidon in additive number theory, and on some related problems', *Journal of the London Mathematical Society*, **1**(4), 212–215, (1941).

[13] Michael R Fellows, Danny Hermelin, Frances Rosamond, and Stéphane Vialette, 'On the parameterized complexity of multiple-interval graph problems', *Theoretical Computer Science*, **410**(1), 53–61, (2009).

[14] Thomas Kalinowski, Nina Narodytska, and Toby Walsh, 'A social welfare optimal sequential allocation procedure', in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI 2013, August 3-9, 2013, Beijing, China*, pp. 227–233, (2013).

[15] Thomas Kalinowski, Nina Narodytska, Toby Walsh, and Lirong Xia, 'Strategic behavior when allocating indivisible goods sequentially', in *Proceedings of the 27th AAAI Conference on Artificial Intelligence, AAAI 2013, July 14-18, 2013, Bellevue, Washington, USA.*, (2013).

[16] Lionel Levine and Katherine E. Stange, 'How to make the most of a shared meal: Plan the last bite first', *The American Mathematical Monthly*, **119**(7), 550–565, (2012).

[17] Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi, 'On approximately fair allocations of indivisible goods', in *Proceedings of the 5th ACM Conference on Electronic Commerce, EC 2004, May 17-20, 2004, New York, NY, USA*, pp. 125–131, (2004).

[18] Yuto Tominaga, Taiki Todo, and Makoto Yokoo, 'Manipulations in two-agent sequential allocation with random sequences', in *Proceedings of the 15th International Conference on Autonomous Agents & Multiagent Systems, AAMAS 2016, May 9-13, 2016, Singapore*, pp. 141–149, (2016).

[19] Toby Walsh, 'Strategic behaviour when allocating indivisible goods', in *Proceedings of the 30th AAAI Conference on Artificial Intelligence, AAAI 2016, February 12-17, 2016, Phoenix, Arizona, USA.*, pp. 4177–4183, (2016).

[20] Mingyu Xiao and Jiaxing Ling, 'Algorithms for manipulating sequential allocation', *To appear in the proceedings of the 34th AAAI Conference on Artificial Intelligence, AAAI 2020. Full version available at CoRR*, **abs/1909.06747**, (2019).