Truthful Mechanisms for Competing Submodular Processes

Allan Borodin, Dept of Computer Science, University of Toronto, bor@cs.toronto.edu
Mark Braverman, Dept of Computer Science, Princeton University, mbraverm@cs.princeton.edu
Brendan Lucier, Microsoft Research, New England, brlucier@microsoft.com
Joel Oren, Dept of Computer Science, University of Toronto, oren@cs.toronto.edu

Motivated by models of competitive influence spread in networks, we study mechanisms for allocating nodes to self-interested agents with negative externalities. For example, a social network provider may wish to allow advertisers to provide special offers to influential individuals. The advertisers benefit in that product adoption may spread through the network. However, a competing product may adversely impact the rate of adoption for any given advertiser.

More generally, there is a ground set U of elements (e.g. nodes representing individuals in a social network) and rational agents (e.g. advertisers) who vie for elements of this set. A valid allocation is a collection of subsets of U – one for each agent. In our applications, the utility of each agent is non-decreasing in its own allocation and non-increasing in its opponents' allocation, and the sum of utilities (the social welfare) is a non-decreasing submodular function. This framework captures many natural models of influence spread, and more generally (set-)monotone processes with negative externalities in which only the total welfare is required to be submodular.

Much of the prior work in modelling competitive influence spread assumes that the players themselves will choose which network nodes to target, and study equilibria of the resulting game. However, in many applications, agents do not have this level of control: instead, the owner of the network chooses the allocation. We therefore consider a mechanism that receives as input a demand from each player, representing the number of elements to allocate, and returns sets respecting those demands with the goal of maximizing social welfare. The introduction of competition raises issues of strategies, where rational agents may strategically underreport their demands. The complicating factor is that while the social welfare function is submodular, the expected utility for each individual agent may undergo complex interactions due to externalities.

In a broad class of such scenarios, the social welfare can be $O(1)$ approximated by an iterative (but not necessarily strategyproof) algorithm that selects an (arbitrary) agent in each iteration and then greedily allocates a node to that agent. Under some natural assumptions about the social welfare and the players' utility functions, we show that a variant of this algorithm with a randomized ordering is strategyproof when there are at least three players. Interestingly, this approach fails to be strategyproof when there are only two advertisers. In the case of two players, we recursively construct a randomized mechanism that is strategyproof and maintains the social welfare approximation of the greedy algorithm. Furthermore, this method also works when the extra assumptions about the social welfare and the players' utilities are lifted.

General Terms: Algorithms, Theory

1. INTRODUCTION

How should social networking platforms, such as Facebook, decide which users should be given special offers from advertisers? It is known that opinions and products can spread through a network as friends influence each others' choices. It is therefore natural to suppose that special offers targeted at consumers that are central in the network topology will have the most impact.

When there is a single advertiser, the manner by which influence spreads in a network is well-studied, with numerous theoretical models having been proposed and analyzed in recent years [Kempe et al. 2003, 2005; Mossel and Roch 2010]. Such models describe a system by which certain initial adopters take a given opinion or product, which in turn causes further adoptions as influence spreads through the network. Generally, the (expectated) final influence is a non-decreasing submodular function of the set of initial adopters. This implies [Nemhauser et al. 1978a] that a natural greedy algorithm can therefore be used to choose initial adopters to approximately maximize expected influence.

While this greedy approach works well in the case of a single advertiser, the model of influence spread becomes significantly more complex when multiple products compete within a network. Tractable models for influence spread in such competitive settings have arisen recently, including the wave propagation model [Carnes et al. 2007], the random delay model [Bharathi et al. 2007], and the OR diffusion model [Borodin et al. 2010]. All of these models have the property that the number of eventual adopters of *any* product is non-decreasing and submodular with respect to the set of initial adopters (e.g. product placements), but the outcome for each individual product can be adversely affected by the inducements for a different product.

The interplay between multiple agents raises game-theoretic issues, as each advertiser is concerned only with the spread of its own influence. This has been previously studied by analyzing the equilibria of a game in which players choose which nodes in the network to target [Bharathi et al. 2007; Carnes et al. 2007; Alon et al. 2010]. However, this approach suffers from the difficulty that it is computationally difficult for a player to determine best-responses in such a game. Moreover, in many applications, the advertisers do not have a level of control that would allow them to target specific nodes. An arguably more natural model would have the advertisers declare the number of special offers they can provide to a central mechanism (such as the social network provider), which then assigns these offers to nodes. Our setting can therefore be viewed as an allocation problem with externalities.

The above problem of distributing nodes among k competing advertisers is thus a special case of the following more general problem (described formally in Section 2). There is a ground set U of elements (e.g. potential adopters). There are k players who vie for elements of this set. A valid allocation is a set $\mathbf{S} = (S_1, \ldots, S_k)$ of subsets of U. The expected utility of agent i is $f_i(\mathbf{S}) = f_i(S_1, \ldots, S_k)$. The social welfare is $f(\mathbf{S}) = \sum_{i=1}^{k} f_i(\mathbf{S})$, which is assumed to be a non-decreasing submodular function in the set[1] $T = \bigcup_{i=1}^{k} S_i$. The mechanism receives as input a demand from each player, representing the number of nodes to allocate, and returns an allocation $\mathbf{S}$ respecting those demands with the goal of maximizing $f(\mathbf{S})$.

In addition to models for influence spread, our general setting also applies to auctions in which agents bid for communication zones (e.g. disks or other regions in the plane). Such an auction can be implemented as a truthful single-minded combinatorial auction if the allocated regions are not permitted to overlap [Babaioff and Blumrosen 2004], but more generally we may allow overlaps which introduce externalities via interference. We discuss this particular application in Appendix A.

We wish to design mechanisms for the above problems that are strategyproof in the sense that rational agents are incentivized to truthfully reveal their demands. The complicating factor is that while the function $f(\mathbf{S})$ is submodular, the expected utility for an agent is negatively impacted by externalities imposed by the allocation to its opponents.

Our Results. Our main results are strategyproof mechanisms for the problem described above which obtain constant factor approximations to the optimal social welfare $f(\cdot)$. For the case of $k > 2$ agents, we show that under two natural assumptions (implicit in [Carnes et al. 2007] and [Bharathi et al. 2007]) about the social welfare and the valuations, there is a mechanism that obtains a $\frac{e}{e-1}$-approximation to the optimal allocation. Specifically, our mechanism first requires that the social welfare is independent of the way in which elements are partitioned amongst the players; i.e. $f(S_1, \ldots, S_k) = f(S'_1, \ldots, S'_k)$ whenever the sets $\bigcup_{i=1}^{k} S_i$ and $\bigcup_{i=1}^{k} S'_i$ are equal. Note that in this special case, our problem can be viewed as a non-decreasing submodular maximization problem subject to a uniform matroid and hence can be $\frac{e}{e-1}$ approximated by the standard greedy algorithm [Nemhauser et al. 1978a]. Second, we require that the utilities of the players are invariant with respect to the manner by which competitor items are partitioned among other players. Under such assumptions, we

[1]Following previous work, many of our results apply to the setting in which allocations must be disjoint, but in some cases (e.g. Section 5) we will allow T to be a multiset.

construct a simple random mechanism is strategyproof (in expectation) and maintains the $\frac{e}{e-1}$ approximation. We also provide evidence in Appendix D that designing a strategyproof mechanism in the more general setting without these assumptions is substantially more difficult. Interestingly, it appears that no such simple method works in the case of only two players, even with our assumptions[2]. Thus, for the case in which there are exactly two players, we must develop a different mechanism.

Our mechanism for two players is based on a novel technique for monotonizing the expected utilities of the agents, using geometric properties of the problem in the two-player case. The construction is based upon a greedy algorithm for submodular function maximization subject to a partition matroid constraint. The standard greedy algorithm (which repeatedly allocates the node that maximizes the marginal welfare) obtains a 2-approximation for this problem, but is not truthful in general. We therefore turn to an alternative algorithm, known as the locally greedy algorithm [Nemhauser et al. 1978a; Goundan and Schulz 2007]. This algorithm repeatedly chooses an (arbitrary) agent in each round, and assigns an element of the base set to that agent in order to maximize the marginal increase to total welfare. We demonstrate by way of examples that this algorithm and its natural randomized variants are also not necessarily truthful: agents may underreport their demands (see Section 4)[3]. However, the locally greedy algorithm has the advantage that the choice of agent in each round is arbitrary; this provides a degree of freedom that can be exploited to design truthful mechanisms. Indeed, for the case of two agents, we show how to recursively construct a distribution over potential allocations returned by locally greedy algorithms, with the property that each agent's expected individual value under this distribution is monotone[4] with respect to the number of initial elements allocated.

In its most general form, our problem need not require that the set of elements allocated to the agents be disjoint. However, many previous models of influence spread do assume disjointness. Our result for 3 or more players (under the mechanism indifference and agent indifference assumptions) holds under the assumption of disjointness: we obtain a strategyproof $e/(e-1)$ approximate mechanism. However, our result for 2 agents without these assumptions does not hold if we require disjointness, and we show that this is essential: if disjointness is required, our mechanism no longer obtains a constant approximation ratio. However, in the case that the agents are *anonymous* (i.e. the utility values are isomorphic under permutation of the allocations) then we show how to modify our two-player mechanism into a strategyproof mechanism for the disjoint case. Specifically, for 2 anonymous agents, we obtain a strategyproof 3 approximation mechanism.

Our mechanisms run in time polynomial in the demands submitted by the agents and in the size of the underlying ground set. This dependence on the demand values is necessary, as the mechanism constructs a solution consisting of sets of this size. Our dependence on the size of the underlying ground set is captured by queries for the element that maximizes the marginal increase in social welfare if it were added to a partial allocation. Given oracle access to queries of this nature, our algorithm would run in time polynomial in the declared demands. Generally speaking, the spread process itself is a randomized process and as in [Kempe et al. 2003, 2005], the oracle can be viewed as providing an element that approximately maximizes the marginal gain by sampling enough trials of the randomized spread process [Kempe et al. 2003, 2005]. Our analysis also holds when such approximate

[2]Notice that the agent indifference property holds vacuously in the two-player case, as there is only one other player.

[3]We assume that each advertiser is constrained in the number of nodes (e.g. special offers) it can provide, and hence can be assumed to never overreport its demand.

[4]We use the word *monotone* in its game-theoretic sense, meaning that a player's outcome is a monotone function of its bid. We distinguish this from the monotonicity of the social welfare function of the mechanism, and use the term *non-decreasing* when referring to the social welfare function.

marginal maximizers are used to implement our underlying greedy algorithm; following the exposition in [Goundan and Schulz 2007], such an approximate marginal maximizer provides an approximation ratio that approaches 2 as the oracle approximation approaches 1. Throughout the paper we will simplify our discussion by assuming that it is possible to find elements that exactly maximize marginal gains in social welfare.

Related Work:. The (non-competitive) problem of maximizing influence in social networks was theoretically modelled by Kempe et al. [Kempe et al. 2003, 2005]. Subsequent papers extended the models studied by Kempe et al. by suggesting similar models in which there are several companies, or technologies, whose influences diffuse throughout the social networks in a competitive manner. Carnes et al. [Carnes et al. 2007] suggested the Wave Propagation model and the Distance Based model, which were based on the Independent Cascade model. Additionally, Dubey et al. [Dubey et al. 2006], Bharathi et al. [Bharathi et al. 2007], Kosta et al. [Kostka et al. 2008], and Apt et al. [Apt and Markakis 2011] also studied various competitive models. The main issue that these models addressed was how to arbitrate ties in each step of the process, determining which technology a node will assume when reached by several technologies at once. The main algorithmic task that these models address is the problem of choosing the optimal set of nodes for a player entering an existing market, in which the competitor's choice of initial nodes is already known. Borodin et al. [Borodin et al. 2010] presented the OR model which proposes a different approach, in which the previously studied, non-competitive diffusion models proceed independently for each technology as a first phase of the process, after which the nodes decide between each technology according to some decision function.

2. PRELIMINARIES

We consider a setting in which there is a ground set $U = \{e_1, \dots, e_n\}$ of n elements, and k players. An allocation is some $(S_1, \dots, S_k) \in 2^U \times \cdots \times 2^U$; that is, an assignment of set[5] S_i to each player i. For the most part we will follow the convention that these sets should be disjoint, though in general our model does not require disjointness. In particular, we consider a setting in which sets need not be disjoint in Section 5.

We are given functions $f_i \colon 2^U \times \cdots 2^U \to \mathbb{R}_{\geq 0}$, denoting the expected values of players $i = 1, \dots, k$, for allocation $(S_1, \dots, S_k)$. We define $f = \sum_{i=1}^{k} f_i$, so that $f(\mathbf{S}) = f(S_i, \mathbf{S_{-i}})$ denotes the total expected welfare of the allocation $(\mathbf{S}) = (S_1, \dots, S_k) = (S_i, \mathbf{S_{-i}})$.

We will require that functions f, and $f_1, \dots, f_k$ satisfy certain properties, motivated by known properties of influence spread models studied in the literature. First, we will assume that f is a submodular non-decreasing function, in the following sense. For any $S_i \subseteq S_i'$, $\mathbf{S_{-i}}$, and $e \in U$, we have

— $f(S_i, \mathbf{S_{-i}}) \leq f(S_i', \mathbf{S_{-i}})$ and
— $f(S_i \cup \{e\}, \mathbf{S_{-i}}) - f(S_i, \mathbf{S_{-i}}) \geq f(S_i' \cup \{e\}, \mathbf{S_{-i}}) - f(S_i', \mathbf{S_{-i}})$.

We will also require that for all $i = 1, \dots, k$, the function f_i be non-decreasing in the allocation to player i, so that $f_i(S_i, \mathbf{S_{-i}}) \leq f_i(S_i', \mathbf{S_{-i}})$ for any $S_i \subseteq S_i'$.

We impose one final model assumption throughout this paper, which we will call the *adverse competition* assumption. Namely, we suppose that f_i will be non-increasing in the allocation to other players, so that for all $j \neq i$, $f_i(S_j, \mathbf{S_{-j}}) \geq f_i(S_j', \mathbf{S_{-j}})$ for any $S_j \subseteq S_j'$. This assumption captures our intuition that, in a competitive influence spread model, the presence of additional adopters for one player can only impede the spread of influence for another player. We discuss the necessity of this assumption in Section 2.1.

We study the following algorithmic problem. Given as input values $b_1, \dots, b_k \geq 0$, we wish to find sets $S_1, \dots, S_k \subseteq U$, with $|S_i| = b_i$, for all $i = 1, \dots, k$, such that $f(S_1, \dots, S_k)$

[5]We emphasize that $S_1, \dots, S_k$ are sets, rather than multisets.

is maximized. We assume we are given oracle access to the functions f, and $f_1, \ldots, f_k$. Note that we impose a "demand satisfaction" condition on the mechanism that each agent is allocated all of his demand. (To this end we will generally assume that $|U| \geq \sum_{i=1}^{k} b_i$; i.e. that there are enough items to allocate).

Suppose that $\mathcal{A}$ is a deterministic algorithm for the above problem, so that $\mathcal{A}(b_1, \ldots, b_k)$ denotes an allocation for any $b_1, \ldots, b_k \geq 0$. We say that $\mathcal{A}$ is *monotone* if, for all bid vectors $\mathbf{b} = (b_1, \ldots, b_k) \in \mathbb{Z}^k_{\geq 0}$, $f_i(\mathcal{A}(b_i, \mathbf{b_{-i}})) \leq f_i(\mathcal{A}(b_i + 1, \mathbf{b_{-i}}))$, for each player $i = 1, \ldots, k$. We extend this definition to randomized algorithms in the natural way, by taking expectations over the outcomes returned by $\mathcal{A}$.

We will assume that each player i has a *type* $\tilde{b_i}$. We say that the utility of player i for allocation $\mathbf{S} = (S_1, \ldots, S_k)$ is

$$u_i(\mathbf{S}) = \begin{cases} f_i(S_i, \mathbf{S_{-i}}) & \text{if } |S_i| \leq \tilde{b_i} \\ -\infty & \text{otherwise} \end{cases}$$

We then say that algorithm $\mathcal{A}$ is *strategyproof* if, for all $\mathbf{b} \in \mathbb{Z}^k_{\geq 0}$ and $b'_i \leq b_i$, $u_i(\mathcal{A}(b'_i, \mathbf{b_{-i}})) \leq u_i(\mathcal{A}(b_i, \mathbf{b_{-i}}))$. In other words, an algorithm is strategyproof if it incentivizes each agent to report its type truthfully.

Going back to the welfare function $f(\cdot)$, we note that the problem of maximizing this function subject to the reported demands can be stated in the framework of maximizing a submodular set-function, subject to a *partition matroid* constraint. An instance of a partition matroid $\mathcal{M} = (E, \mathcal{F})$ is given by a union of disjoint sets $E = \bigcup_{i=1,\ldots,k} E_i$, and a set of corresponding cardinality constraints $d_1, \ldots, d_k$. A set X is in $\mathcal{F}$, i.e. is *independent*, if $|X \cap E_i| \leq d_i$, for all $1 \leq i \leq k$. That is, an independent set is formed by taking no more than the prescribed size constraint for each of the sets. The optimization task would therefore be to find an independent set that maximizes a non-decreasing and submodular set-function $g : \mathcal{F} \to \mathbb{R}_{\geq 0}$. Our problem falls into this framework by setting the ground set to be $U \times \{1, \ldots, k\}$, the cardinality constraints $d_i = b_i$, for all $i = 1, \ldots, k$ and setting the objective function to be the social welfare:

$$g(X) = f(\mathbf{S}), \text{ where } X = \bigcup_{i=1}^{k} (S_i \times \{i\}). \tag{1}$$

We note, however, that this formulation does not apply if the allocated sets are required to be disjoint. Indeed, the addition of disjointness causes our constraint to no longer take the form of a matroid, an issue which will be addressed in Section 6. Also note that this alternative definition of our setting conforms to the single-parameter convention of submodular set-functions. However, we will mostly refer to the former formulation of the problem for clarity and succinctness.

As a result of this correspondence with the framework of partition matroids, we will be interested in a particular greedy algorithm for this algorithmic problem, known as a *locally greedy* algorithm, studied in [Nemhauser et al. 1978b], which was subsequently extended in [Goundan and Schulz 2007].

The algorithm proceeds by fixing some arbitrary permutation of the multiset composed of b_i i's for each player i. It then iteratively builds the allocation $\mathbf{S}$ where, on iteration j, it chooses $u \in \arg\max_c \{f(S_i \cup \{c\}, \mathbf{S_{-i}}) - f(S_i, \mathbf{S_{-i}})\}$ and adds u to S_i, where i is the jth element of the permutation. Regardless of the permutation selected, this algorithm is guaranteed to obtain a 2-approximation to the optimal allocation subject to the given cardinality constraints [Nemhauser et al. 1978b; Goundan and Schulz 2007].

2.1. The Adverse Competition Assumption

As mentioned, we place a restriction on the valuation functions: adding an item to one player's set cannot improve the outcome of its competitors. Although the greedy approximation algorithm does not, in general, require this property in order to guarantee a constant approximation ratio for the social welfare, it is tempting to consider what would happen if one were to lift this assumption about the valuations.

A simple example shows that the assumption of adverse competition is necessary for truthfulness. Consider the following two-player setting. The ground set is composed of two items: u_1, which contributes a value of 1 to the receiving player and a value of N to her competitor (who did not receive u_1), and item u_2 which gives both players a value of 1.

Now, consider the outcome of any mechanism when the bid profile is $(1, 1)$. Without loss of generality, one player, say player A, will receive u_1, while the other player will get u_2. The valuations would therefore be 2 and $N + 1$ for players A and B, respectively. In that case, player A would prefer to lower her bid to 0, which would guarantee her a valuation of N (player B would have to get u_1, as otherwise the approximation ratio of the social welfare is unbounded as N grows).

We conclude that unless the competition assumption holds, no strategyproof mechanism can, in general, obtain a bounded approximation ratio to the optimal social welfare. Also, although the discussion above refers to deterministic allocations, the same argument can be made for randomized allocations as well.

3. A TRUTHFUL MECHANISM FOR MORE THAN TWO PLAYERS

We begin our study of methods for obtaining strategyproof mechanisms by considering a framework in which we place additional restrictions on the social welfare function, and the individual utility functions. These additional restrictions are satisfied by many models of influence spread considered in the literature, as we discuss below. We show that in such a setting, there is a natural strategyproof mechanism when there are at least three players. In fact, it turns out that having three or more players in such a setting allows for a much simpler mechanism than the mechanism for the case of only two players.

Assumption 1: Mechanism Indifference. We will assume that $f(\mathbf{S}) = f(\mathbf{S}')$ whenever the sets $\bigcup_i S_i$ and $\bigcup_i S'_i$ are equal. That is, social welfare does not depend on the manner in which allocated items are partitioned between the agents. We will call this the *Mechanism Indifference* (MeI) assumption.

If assumption 1 holds, then we can imagine a greedy algorithm that chooses which items to add to the set $\bigcup_i S_i$ one at a time to greedily maximize the social welfare. By assumption 1, the welfare does not depend on how these items are divided among the players. This greedy algorithm generates a certain social welfare whenever the sum of budgets is t; write $w(t)$ for this welfare. Note that $w(0), w(1), \ldots$ is a concave non-decreasing sequence.

Assumption 2: Agent Indifference. We will assume that $f_i(S_i, \mathbf{S}_{-i}) = f_i(S_i, \mathbf{S}'_{-i})$ whenever sets $\bigcup_{j \neq i} S_j$ and $\bigcup_{j \neq i} S'_j$ are equal. That is, each agent's utility depends on the set of items allocated to the other players, but not on how the items are partitioned among those players. We will call this the *Agent Indifference* (AgI) assumption. Notice that in the two-players case, this assumption is essentially vacuous.

We note that the models for competitive influence spread proposed by Carnes et al. [Carnes et al. 2007] and Bharathi et al. [Bharathi et al. 2007] are based on a cascade model of influence spread, and satisfy both the MeI and AgI assumptions. Similarly, if we restrict the OR model in [Borodin et al. 2010] so that the underlying spread process is a cascade (and not a threshold) process and agents are anonymous (a restriction that will be defined in Section 6), as assumed in the Carnes et al models, then this special case of the OR model also satisfies MeI and AgI.

3.1. The uniform random greedy mechanism

Consider Algorithm 1, which we refer to as the uniform random greedy mechanism. This mechanism proceeds by first greedily selecting which elements of the ground set to allocate. It then chooses an ordering of the players' bids uniformly at random from the set of all possible orderings, then assigns the selected elements to the players in this randomly chosen order. The MeI assumption implies that the random greedy mechanism obtains a constant

Algorithm 1: Uniform Random Greedy Mechanism

Input: Ground set $U = \{e_1, \ldots, e_m\}$, budget profile $\mathbf{b}$
Output: An allocation profile $\mathbf{S}$

1 Initialize: $S_i \leftarrow \emptyset, i \leftarrow 0, j \leftarrow 0, I \leftarrow \emptyset, t \leftarrow \sum_i b_i$;
`/* Choose elements to assign. */`
2 **while** $i < t$ **do**
3 $\quad u_i \leftarrow argmax_{c \in U}\{f(I \cup \{c\}) - f(I)\}$;
4 $\quad I \leftarrow I \cup \{u_i\}$; $i \leftarrow i + 1$;
5 **end**
`/* Partition elements of` I. `*/`
6 $\Gamma \leftarrow \{\beta : [t] \rightarrow [k]$ s.t. $|\beta^{-1}(i)| = b_i$ for all $i\}$;
7 Choose $\beta \in \Gamma$ uniformly at random ;
8 **while** $j < t$ **do**
9 $\quad S_{\beta(j)} \leftarrow S_{\beta(j)} \cup \{u_j\}$;
10 $\quad j \leftarrow j + 1$;
11 **end**

factor approximation to the optimal social welfare. We now claim that, under the MeI and AgI assumptions, this mechanism is strategyproof as long as there are at least 3 players.

THEOREM 3.1. *If there are $k \geq 3$ players and the AgI and MeI assumptions hold, then Algorithm 1 is a strategyproof mechanism. Furthermore, Algorithm 1 approximates the social welfare to within a factor of $\frac{e}{e-1}$ from the optimum.*

PROOF. As before, notice that lines 2–5 are an implementation of the standard greedy algorithm for maximizing a non-decreasing, submodular set-function subject to a uniform matroid constraint, as described in [Nemhauser et al. 1978b; Goundan and Schulz 2007], and hence gives the specified approximation ratio.

Next, we show that Algorithm 1 is strategyproof. Fix bid profile $\mathbf{b}$ and let $t = \sum_i b_i$. Let I be the union of all allocations made by Algorithm 1 on bid profile $\mathbf{b}$; note that I depends only on t. Furthermore, each agent i will be allocated a uniformly random subset of I of size b_i. Thus, the expected utility of agent i can be expressed as a function of b_i and t. We can therefore write $w^i(b,t)$ for the expected utility of agent i when $b_i = b$ and $\sum_j b_j = t$ (recall that we let $w(t)$ denote the total social welfare when $\sum_i b_i = t$).

We now claim that $w^i(b,t) = \frac{b}{t} w(t)$ for all i and all $0 \leq b \leq t$. Note that this implies the desired result, since if our claim is true then for all i and all $0 \leq b \leq t$ we will have

$$w^i(b,t) = \frac{b}{t} w(t) \leq \frac{b+1}{t+1} w(t+1) = w^i(b+1, t+1)$$

which implies the required monotonicity condition. It now remains to prove this claim.

Note first that the adverse competition assumption implies that $w^i(0,t) \leq w^i(0,0) = 0$ for all i and t. We next show that $w^i(1,t) = w^j(1,t)$ for all i, j, and $t \geq 1$. If $t = 1$ then this follows from the MeI assumption. So take $t \geq 2$ and pick any three agents i, j, and ℓ. Then, by the AgI assumption, we have

$$w^i(1,t) = w(t) - w^\ell(t-1,t) = w^j(1,t).$$

We next show that $w^i(b,t) = w^i(1,t) + w^i(b-1,t)$ for all i, all $b \geq 2$, and all $t \geq b$. Pick any three agents i, j, and ℓ, any $b \geq 2$, and any $t \geq b$. We then have, by the AgI assumption,

$$\begin{aligned} w^i(b,t) &= w(t) - w^\ell(t-b,t) \\ &= w(t) - [w(t) - w^i(b-1,t) - w^j(1,t)] \\ &= w^i(b-1,t) + w^j(1,t) \\ &= w^i(b-1,t) + w^i(1,t). \end{aligned}$$

It then follows by simple induction that $w^i(b,t) = bw^i(1,t)$ for all $1 \leq b \leq t$. But now note that $w(t) = w^i(1,t) + w^j(t-1,t) = tw^i(1,t)$, and hence $w^i(1,t) = \frac{1}{t}w(t)$ and therefore $w^i(b,t) = \frac{b}{t}w(t)$ for all $0 \leq b \leq t$. This completes the proof of the claim. □

Note that the proof of Theorem 3.1 makes crucial use of the fact that there are at least three players. Indeed, the OR model satisfies the MeI and AgI assumptions when $k = 2$, and we have examples in the OR model for which the random greedy algorithm is not strategyproof. Thus, for the case $k = 2$, it is necessary to develop a novel strategyproof mechanism (such as the one we present in our paper).

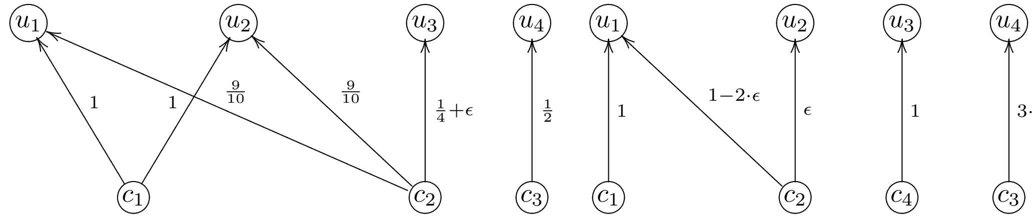
4. COUNTER EXAMPLES WHEN THERE ARE TWO AGENTS

In contrast to the results for $k > 2$ agents, Algorithm 1 is no longer strategyproof when there are only two agents. Furthermore, we demonstrate that the locally greedy algorithm due to Nemhauser et al [Nemhauser et al. 1978a] (see also Goundan and Schultz [Goundan and Schulz 2007]) and Section 2) is not, in general, strategyproof for other natural methods for choosing the ordering of the two agents. We recall that the Agent Indifference assumption is vacuous for 2 agents. To clarify the context when there are only two agents, we refer to them as agent A and agent B and their utilites as f_A and f_B respectively. We give examples of a set U and functions f_A and f_B (satisfying the conditions of our model) such that natural greedy algorithms for choosing sets S and T result in non-monotonicities. Our examples satisfy the MeI assumption and as previously noted, the AgI assumption is vacuous for two agents. Our examples will all easily extend to the case of $k > 2$ agents noting that these examples will then not satisfy agent indifference if there are more than two agents.

4.1. The OR model

We will consider examples of a special case of the OR model for influence spread, as studied in [Borodin et al. 2010]. Let $G = (V, E)$ be a graph with fractional edge-weights $p : E \rightarrow [0,1]$, vertex weights w_v for each $v \in V$, and sets $I_A, I_B \subseteq V$ of "initial adopters" allocated to each player. We use vertex weights for clarity in our examples; in Appendix B we show how to modify the examples given in this section to be unweighted. The process then unfolds in discrete steps. For each $u_A \in I_A$ and v_A such that $(u_A, v_A) \in E$, u_A, once infected, will have a single chance to "infect" v_A with probability $w(u_A, v_A)$. Define the same, single-step process for the nodes in I_B, and let O_A and O_B be the nodes infected by nodes in I_A and I_B, respectively. Note that the infection process defined for each individual player is an instance of the Independent Cascade model as studied by Kempe et al. [Kempe et al. 2003]. Finally, nodes that are contained in $O_A \backslash O_B$ will be assigned to player A, nodes in $O_B \backslash O_A$ will be assigned to B, and any nodes in $O_A \cap O_B$ will be assigned to one player or the other by flipping a fair coin.

In our examples, we consider two identical players each having utility equal to the weight of the final set of nodes assigned by the spread process. It can be easily verified that both



(a) The counter-example for the deterministic mechanism with a dictatorship ordering. The initial budget for both players is 1.

(b) The counter-example for the deterministic algorithm under a Round Robin ordering. The initial budgets for players A and B are 1 and 2, respectively.

Fig. 1: Counter-examples for the mechanism under the deterministic dictatorship and Round Robin orderings. In both case, we set the weights $w_{c_i} = \epsilon$ and $w_{u_i} = 1$, for all $1 \leq i \leq 4$. $0 < \epsilon < \frac{1}{8}$

the expected social welfare (total weight of influenced nodes) and the expected individual values (fixing the other player's allocation) are submodular set-functions.

4.2. Deterministic greedy algorithms that are not strategyproof

We demonstrate that the more obvious deterministic orderings for the greedy algorithm fail. First, consider the "dictatorship" ordering, in which (without loss of generality by symmetry) player A is first allocated nodes according to his budget, and only then player B is allocated nodes. Our example showing non-truthfulness also applies to an ordering that would always select the player having the largest current unsatisfied budget breaking ties (again without loss of generality by symmetry) in favor of player A. Consider the graph depicted in Figure 1a. When player A bids 1 and player B bids 1 as well, the algorithm will allocate c_1 to player A, as it contributes the maximal marginal gain of the social welfare, and will allocate c_3 to player B. The value of the allocation of player A is 2.

However, notice that if player A increases its bid to 2, the mechanism will allocate nodes c_1 and c_3 to player A, and allocate c_2 to B. In this case player A receives an extra value of $\frac{1}{2}$ from node c_3, but the allocation of c_2 to B will "pollute" player A's value from c_1: he will receive nodes u_1 and u_2 with probability $\frac{1}{10} + \frac{1}{2} \cdot \frac{9}{10} = \frac{11}{20}$. Thus the total expected value for player A is only $\frac{16}{10}$, and hence the algorithm is non-monotone in the bid of player A.

Next, consider the Round Robin ordering, in which the mechanism alternates between allocating a node to player A and to player B. Our example here also applies to the case when the mechanism always chooses the player having the smallest current unsatisfied budget breaking ties in favor of player A. Consider the instance given in Figure 1b. When the bids of players A and B are 1 and 2, respectively, the algorithm will first allocate c_1 to player A, and then it will subsequently allocate nodes c_3 and c_4 to player B, which results in a payoff of 1 for player A. If player A were to increases his bid to 2, then the mechanism would allocate nodes c_1 and c_3 to player A, and nodes c_2 and c_4 to player B, for a payoff of $3 \cdot \epsilon + 2 \cdot \epsilon + (1 - 2 \cdot \epsilon) \cdot \frac{1}{2} = \frac{1}{2} + 4 \cdot \epsilon < 1$ (since $0 < \epsilon < \frac{1}{8}$). Therefore, the monotonicity is violated w.r.t. the payoff of player A.

4.3. The uniform random greedy algorithm is not strategyproof

In light of Algorithm 1, one might wonder if a simple randomized algorithm might also be strategyproof for two agents. Unfortunately, it is not always the case that such an approach would yield a strategyproof allocation algorithm. Consider the example given in Figure 2. We note that for this example, Algorithm 1 is equivalent to first choosing a random order of allocation (e.g. choosing all possible permutations satisfying agent demands with equal probability) and then allocating greedily. The greedy algorithm will allocate one of c_2, c_3, c_4

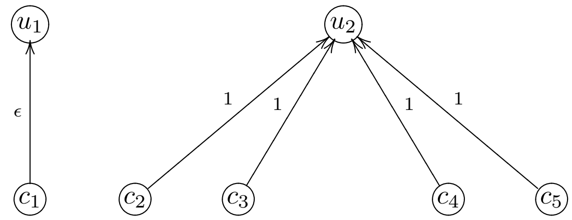


Fig. 2: The counterexample for the mechanism that allocated according to a random ordering of the turns ($0 < \epsilon \ll 1$). $w_{c_i} = \epsilon, i = 1, \ldots, 5$, $w_{u_i} = 1, i = 1, 2$

and c_5 to one of the players, then allocate c_1, and then any remaining nodes.

Let player A's budget be 3 and player B's budget be 1. In this case, with probability $\frac{1}{4}$, player B will be allocated c_1 (i.e. when B's allocation occurs second), in which case player A's expected value would be 1. Also, with probability $\frac{3}{4}$, player B will be allocated one of $\{c_2, c_3, c_4, c_5\}$, in which case player A's expected outcome would be $\frac{1}{2} + \epsilon$. In total, player A's expected payoff will be $\frac{5}{8} + \frac{3}{4}\epsilon$.

If player A were to increase his budget to 4, then with probability $\frac{1}{5}$ player B will be allocated c_1, in which case player A's outcome will be 1. On the other hand, player A's expected payoff will be $\frac{1}{2} + \epsilon$ if B is allocated one of $\{c_2, c_3, c_4, c_5\}$, which occurs with probability $\frac{4}{5}$. In total player A's expected outcome will be $\frac{3}{5} + \frac{4}{5}\epsilon < \frac{5}{8} + \frac{3}{4}\epsilon$, implying that this algorithm is non-monotone.

5. THE TWO PLAYERS CASE: A STRATEGYPROOF MECHANISM

In this section we describe our mechanism for allocating nodes in a competitive submodular process for the case of only two players. As opposed to our discussion on the $k > 2$ agent case, our mechanism works without the MeI and AgI assumptions. This mechanism is based on the local greedy algorithm described in Section 2. In this Section we will focus on cases in which the allocations to the two agents need not be disjoint. In Section 6 we show that our mechanism can be modified in a straightforward way to handle disjointness constraints when agents are "anonymous."

Recall that without strategyproofness concerns we could maximize the value of our submodular function by way of a straightforward greedy algorithm. This greedy algorithm proceeds by iteratively allocating a single node to some player in order to maximize the marginal increase in total influence, subject to the given budget constraints. However, as we demonstrated in Section 4, this algorithm is not necessarily monotone and could incentivize agents to misreport their budgets.

One nice property of the greedy algorithm described in Section 2 is that the guarantee on the social welfare (i.e. the worst case approximation ratio of 2) holds even if we arbitrarily fix the order in which allocations are made to players A and B. Thus, given a particular pair of budgets (a, b), our approach will be to randomize over possible orderings in which to allocate to the two agents, and then apply the greedy algorithm to whichever permutation we choose. The key to the algorithm will be the manner in which we choose the distribution to randomize over, which will depend on the budgets we are given. Indeed, it is not even clear a priori that distributions exist that simultaneously monotonize the expected allocation for both players. Our main technical contribution is a proof that such distributions do exist, and moreover can be explicitly constructed in polynomial time.

The idea behind our construction, at a high level, is as follows. We will construct the distribution for use with budgets (a, b) recursively. Writing $t = a + b$, we first generate distributions for the case $t = 1$ (which are trivial), followed by $t = 2$, etc. To construct the distribution for demands (a, b), we consider the following thought experiment. We will choose an ordering in one of two ways. Either we choose a permutation according to the distribution for budget pair $(a-1, b)$ and then append a final allocation to A, or else choose a permutation according to the distribution for budget pair $(a, b-1)$ and append an allocation to player B. If we choose the former option with some probability α, and the latter with probability $1-\alpha$, this defines a probability distribution for budget pair (a, b).

What we will show is that, assuming our distributions are constructed to adhere to certain invariants, we can choose this α such that the resulting randomized algorithm (i.e. the greedy algorithm applied to permutations drawn from the constructed distributions) will be monotone. That is, the expected influence of player A under the distribution for (a, b) is at least that of the distribution for $(a-1, b)$, and similarly for player B. We note that the existence of such an α is not guaranteed in general; we will actually need to prove that our constructed distributions satisfy an additional "cross-monotonicity" property in order to guarantee that such an α exists.

One problem with the above technique is that it does not bound the size of the support of the distributions over permutations being considered. In general there will be exponentially many possible permutations to randomize over, which leads to exponential computational complexity when attempting to compute the probability α. One might attempt to overcome such issues by sampling and estimating the quantities needed to compute α, but this introduces the possibility of non-monotonicities due to sampling error, which we would like to avoid. We demonstrate that each distribution we construct can be "pruned" so that its support contains at most three permutations, while still retaining its monotonicity properties. In this way, we guarantee that our recursive process requires only polynomially many queries (of the expected welfare generated by a given allocation) in order to choose a permutation.

5.1. The Allocation Algorithm

Our algorithm will proceed by choosing a distribution over orders in which nodes are allocated to the two players. This will be stored in a matrix M, where $M[a, b]$ contains a distribution over sequences $(y_1, \ldots, y_t) \in \{A, B\}^{a+b}$, containing a 'A's and b 'B's. We then choose a sequence from distribution $M[a, b]$ and greedily construct a final allocation with respect to that ordering. We begin by describing the manner in which the allocation is made, given the distribution over orderings. The algorithm is given as Algorithm 2.

An important property of the allocation algorithm that we will require for our analysis is that, given a sequence drawn from distribution $M[a, b]$, the allocation is chosen myopically. That is, items are chosen for the players in the order dictated by the given sequence, independent of subsequent allocations. We will use this property to construct the distribution $M[a, b]$, which will be tailored to the specific algorithm to ensure strategyproofness. We note that this technique could be applied to *any* allocation algorithm with this property; we will make use of this observation in Section 6.

Recall that the approximation guarantee for the greedy allocation does not depend on the order of assignment implemented in lines 3-11, so that the allocation returned by the algorithm will be a 2-approximation to the optimal total influence regardless of the permutation chosen on line 2. It remains only to demonstrate that we can construct our distributions in such a way that the expected payoff to each player is monotone increasing in his bid.

5.2. Constructing matrix M

We describe the procedure *ConstructDistributions*, used in Algorithm 2, to generate distributions over orderings of assignments to players A and B. We will build table $M[\cdot, \cdot]$ recursively, where $M[a, b]$ describes the distribution corresponding to budgets a and b. Our procedure will terminate when the required entry has been constructed.

We think of $M[a, b]$ as a distribution over sequences of the form $(y_1, \ldots, y_{a+b})$, where $y_i \in \{A, B\}$. Note, however, that for any given sequence, the corresponding allocation is

Algorithm 2: Allocation Mechanism

Input: Ground set $U = \{e_1, \ldots, \ldots, e_n\}$, budgets a, b for players A and B, respectively
Output: An allocation $I_A, I_B \subseteq U$ for the two players

`/* Build permutation table. */`
$M \leftarrow ConstructDistributions(a, b)$;
`/*` $M[a, b]$ `will be a distribution over sequences` $(y_1, \ldots, y_{a+b}) \in \{A, B\}^{a+b}$ `*/`
Choose $(y_1, \ldots, y_{a+b})$ from distribution $M[i, j]$;
for $i = 1 \ldots a + b$ **do**
  if $y_i =' A'$ **then**
    $u \leftarrow argmax_{c \in U}\{f(I_A \cup \{c\}, I_B) - f(I_A, I_B)\}$;
    $I_A \leftarrow I_A \cup \{u\}$;
  else
    $u \leftarrow argmax_{c \in U}\{f(I_A, I_B \cup \{c\}) - f(I_A, I_B)\}$;
    $I_B \leftarrow I_B \cup \{u\}$;
  end
end

determined since the greedy algorithm applied in Algorithm 2 is deterministic. We can therefore also think of $M[a, b]$ as a distribution over allocations, and in what follows we will often refer to allocations drawn from $M[a, b]$ without further comment.

Note that $M[0, b]$ must be the distribution that assigns probability 1 to the sequence $(B, B, \ldots, B)$, and similarly $M[a, 0]$ assigns probability 1 to $(A, A, \ldots, A)$. We will construct the remaining entries of the table $M[a, b]$ in increasing order of $a + b$.

Before describing the recursive procedure for filling the table, we provide some notation. Given table M, we will write $w^A(a, b)$ for the expected value of agent A under the distribution of allocations returned by $M[a, b]$. Similarly, $w^B(a, b)$ will be the expected value of agent B, and $w(a, b) = w^A(a, b) + w^B(a, b)$ is the expected total welfare. For notational convenience, we will set $w^A(a, b) = w^B(a, b) = 0$ if $a < 0$ or $b < 0$.

We will construct M so that the following invariants hold for all $a > 0$ and $b > 0$:

(1) $w^A(a, b) \geq w^A(a - 1, b + 1)$.
(2) $w^A(a, b) \geq w^A(a - 1, b)$.
(3) $w^B(a, b) \geq w^B(a, b - 1)$.
(4) The support of $M[a, b]$ contains at most 3 sequences.

The first invariant is a type of cross-monotonicity property, which will help us to construct the entries of matrix M. The second two desiderata capture the monotonicity properties we require of our algorithm. Note that if M satisfies these properties, then Algorithm 2 will be monotone and hence strategyproof. The final property limits the complexity of constructing and sampling from $M[a, b]$, implying that Algorithm 2 runs in polynomial time.

We now describe the way in which we construct distribution $M[a, b]$, given distributions $M[a', b']$ for all $a' + b' < a + b$. We first consider two distributions: the first selects a sequence according to $M[a - 1, b]$ and appends an 'A', and the second selects a sequence according to $M[a, b - 1]$ and appends a 'B'. Call these two distributions D_1 and D_2, respectively. What we would like to do is find some α, $0 \leq \alpha \leq 1$, such that if we choose from distribution D_1 with probability α and distribution D_2 with probability $1 - \alpha$, then the resulting combined distribution (for $M[a, b]$) will satisfy $w^A(a, b) \geq w^A(a - 1, b)$ and $w^B(a, b) \geq w^B(a, b - 1)$. Of course, this combined distribution may have support of size up to 6 (3 from D_1 and 3 from D_2) but we will show that it can be pruned to a distribution with the same expected influence for agents A and B, with at most 3 permutations in its support.

Our main technical lemma, Lemma 5.1, demonstrates that an appropriate value of α, as described in the process sketched above, is guaranteed to exist and can be found efficiently. Before stating the Lemma we introduce some helpful notation. Write $\Delta^{\oplus B}(a,b) = w(a,b) - w(a,b-1)$. That is, $\Delta^{\oplus B}(a,b)$ is the marginal gain in total welfare when agent B increases his bid from $b-1$ to b, given matrix M. Our main Lemma is now as follows:

LEMMA 5.1. *It is possible to construct table M in such a way that the following properties hold for all $a + b \geq 1$:*

(1) $w^A(a,b) \geq w^A(a-1,b+1)$
(2) $w^A(a,b) \geq w^A(a-1,b)$
(3) $w^A(a,b) \leq w^A(a,b-1) + \Delta^{\oplus B}(a,b)$

Furthermore, the entries of M can be computed in polynomial time.

Notice that condition 3 in Lemma 5.1 implies that player B's valuation is monotone increasing with his bid:

$$\begin{aligned} w^A(a,b-1) &\geq w^A(a,b) - \Delta^{\oplus B}(a,b) \\ &= w^A(a,b) - [w(a,b) - w(a,b-1)] \\ &= w^A(a,b) - [(w^A(a,b) + w^B(a,b)) - (w^A(a,b-1) + w^B(a,b-1))] \\ &= w^A(a,b-1) + w^B(a,b-1) - w^B(a,b) \\ &\Rightarrow w^B(a,b) \geq w^B(a,b-1) \end{aligned} \tag{2}$$

PROOF. We will proceed by induction on $t = a + b$. The result is trivial for $t = 1$.

Given a and b with $a+b = t$, we will generate distribution $M[a,b]$ by constructing a value α, then with probability α we choose from the distribution of sequences (i.e. specifying an order of allocations) $M[a-1,b]$ and append A, or else with probability $1-\alpha$ we choose from the distribution $M[a,b-1]$ and append B. We must show the existance of some α value such that the three condition required by Lemma 5.1 will hold.

Conditions 2 and 3 of the Lemma describe an interval in which the value $w^A(a,b)$ must fall. We will call this interval $I_m^{a,b}$. That is,

$$I_m^{a,b} = [w^A(a-1,b), w^A(a,b-1) + \Delta^{\oplus B}(a,b)].$$

Claim 1 shows that this interval is non-empty.

CLAIM 1. $w^A(a-1,b) \leq w^A(a,b-1) + \Delta^{\oplus B}(a,b)$.

PROOF. This follows by induction applied to condition 1 of the Lemma, which implies $w^A(a-1,b) \leq w^A(a,b-1) \leq w^A(a,b-1) + \Delta^{\oplus B}(a,b)$. $\square$

Let W_1^A (respectively, W_1^B) denote the expected payoff of player A (respectively, player B) if we let $\alpha = 1$. That is, W_1^A is the expected influence of player A from the allocation that results if we select a permutation from $M[a-1,b]$ and append A, then use this permutation when applying our greedy algorithm. We define W_0^A and W_0^B similarly for $\alpha = 0$. The following claim follows from the fact that the players' valuations are non-decreasing, and from the adverse competition assumption.

CLAIM 2. $W_1^A \geq w^A(a-1,b)$ *and* $W_0^A \leq w^A(a,b-1)$.

PROOF. The first part of the claim follows because, for each fixed ordering in the support of $M[a-1,b]$, appending an A to that ordering can only increse the welfare of agent A.

Likewise, the second part of the claim follows because, for each ordering in the support of $M[a, b-1]$, appending a B can only decrease the welfare of agent A. $\square$

We think of W_1^A and W_0^A as the influence for agent A for distributions that we can construct. We let $I_c^{a,b}$ denote the interval between W_1^A and W_0^A. Note that we do not know, a priori, which of W_1^A or W_0^A is greater. Using Claim 2 it can be shown that

Claim 3. $I_m^{a,b} \cap I_c^{a,b} \neq \emptyset$

Proof. It cannot be that $I_c^{a,b}$ lies entirely above $I_m^{a,b}$, since $W_0^A \leq w^A(a, b-1) \leq w^A(a, b-1) + \Delta^{\oplus B}(a, b)$. Also, it cannot be that $I_c^{a,b}$ lies entirely below $I_m^{a,b}$, since $W_1^A \geq w^A(a-1, b)$. Thus $I_m^{a,b} \cap I_c^{a,b} \neq \emptyset$. $\square$

We can therefore write $I^{a,b} = I_m^{a,b} \cap I_c^{a,b}$. Note that any point in $I^{a,b}$ corresponds to a distribution we can construct for $M[a, b]$, which will satisfy conditions 2 and 3 of our Lemma. It remains to show that we can choose this point so that condition 1 of Lemma 5.1 will also be satisfied. Our claim is that if we always choose α so that $w^A(a, b)$ is the minimum endpoint of $I^{a,b}$, then condition 1 will be satisfied.

With the above in mind, we will set

$$\alpha = \underset{\alpha \in [0,1]}{\arg\min} \{\alpha W_1^A + (1-\alpha) W_0^A \in I\} \tag{3}$$

Note that if we use this value of α to randomize between appending A to a permutation drawn according to $M[a-1, b]$ and appending B to a permutation drawn according to $M[a, b-1]$, then the resulting value of $w^A(a, b)$ will indeed be $\min I^{a,b}$.

We will define $M[a', b']$ in this way for all $a' + b' = t$. We now argue that this choice satisfies condition 1 of our Lemma.

Claim 4. *If $a \geq 1$ then $w^A(a, b) \geq w^A(a-1, b+1)$.*

Proof. Note first that $w^A(a, b) \geq w^A(a-1, b)$, since $w^A(a, b) \in I_m^{a,b}$. Consider now the value of $w^A(a-1, b+1)$, which is the minimum of $I_c^{a-1,b+1} \cap I_m^{a-1,b+1}$. We will now bound the value of $w^A(a-1, b+1)$, by providing an upper bound on both the minimal endpoint of $I_c^{a-1,b+1}$ and the minimal endpoint of $I_m^{a-1,b+1}$.

For budgets $(a-1, b+1)$, the lower endpoint of $I_m^{a-1,b+1}$ is $w^A(a-2, b+1)$. On the other hand, $I_c^{a-1,b+1}$ contains point W_0^A, which is the influence to player A when we choose a permutation according to $w^A(a-1, b)$ and append a 'B'. However, since allocating an additional item to player B in any fixed allocation can only degrade player A's payoff, it must be that $W_0^A \leq w^A(a-1, b)$.

Thus the lower endpoint of $I_m^{a-1,b+1} \cap I_c^{a-1,b+1}$ is at most $\max\{w^A(a-2, b+1), w^A(a-1, b)\}$. But $w^A(a-2, b+1) \leq w^A(a-1, b)$ by induction (using condition 1 of Lemma 5.1).

We therefore conclude $w^A(a-1, b+1) \leq \max\{w^A(a-2, b+1), w^A(a-1, b)\} \leq w^A(a-1, b) \leq w^A(a, b)$, as required. $\square$

We have shown that table M can be filled with distributions that satisfy the conditions of Lemma 5.1. It remains to discuss the complexity of computing the entries of M. In the argument above, we do not bound the size of the support of the distributions in M. We now wish to modify the argument to show that the number of permutations required for each table entry $M[a, b]$ can be limited to only three, again by induction on $t = a + b$.

Consider the distribution constructed for $M[a, b]$. The support of this distribution has size at most 6: the three permutations in the support of $M[a-1, b]$ with A appended, plus the three permutations in the support of $M[a, b-1]$ with B appended. Each of these six permutations implies an allocation, say $(S_1, T_1), \ldots, (S_6, T_6)$. For each of these allocations, we can consider the two-dimensional point $(f_A(S_i, T_i), f_B(S_i, T_i))$ representing the welfare to A and B for the given allocation. We can interpret our construction of $M[a, b]$ as implementing a point $(w^A(a,b), w^B(a,b))$ with certain properties, such that this point lies in the convex hull of the six points $(f_A(S_1,T_1), f_B(S_1,T_1)), \ldots, (f_A(S_6,T_6), f_B(S_6,T_6))$.

We now use the following well-known theorem [Rockafellar 1996]:

THEOREM 5.1 (CARATHÉODORY). *Given a set $V \subset \mathbb{R}^n$ and a point $p \in ConvV$ — the convex hull of V, there exists a subset $A \subset V$ such that $|A| \leq n+1$ and $p \in ConvA$.*

It must therefore be that our point $(w^A(a,b), w^B(a,b))$ lies in the convex hull of at most three of the points $(f_A(S_1,T_1), f_B(S_1,T_1)), \ldots, (f_A(S_6,T_6), f_B(S_6,T_6))$. It follows that there exists a distribution with a support that consists of three of the six permutations corresponding to (a,b). Finding this distribution can be done in constant time by considering $\binom{6}{3}$ sets of three allocations.[6] We can therefore construct $M[a,b]$ as a distribution over at most 3 permutations, concluding the proof of Lemma 5.1.

The proof of Lemma 5.1 is constructive: it implies a recursive method for constructing the table M of distributions. That is, the procedure *ConstructDistributions* from Algorithm 2 (with input (a,b)) will procede by filling table M in increasing order of t, up to $a+b$, by choosing the value of α for each table entry as in the proof of Lemma 5.1, then storing the implied distribution over three permutations. Note that we can explicitly store the allocations corresponding to the permutations in the table, making it simple to compute the submodular function values needed to determine α (which is store as well).

We conclude, given this implementation of *ConstructDistributions*, that Algorithm 2 is a polytime strategyproof 2-approximation to the 2-player influence maximization problem.

6. DISJOINT ALLOCATIONS

We now show how to modify the mechanism from Section 5 so as to ensure disjoint allocations. Recall that our general strategy in the non-disjoint case was to first fix a myopic allocation algorithm and then construct a strategyproof-inducing distribution over player orderings, for that algorithm. We noted that this method can be applied to any myopic allocation with a social welfare guarantee that does not depend on the chosen order of players. That is, while different algorithms will generate different approximation factors, our strategyproof construction depends only on the myopic nature of the allocation.

In this section we will apply the same method, but will use an algorithm that generates disjoint allocations. However, we note that when this disjointness constraint is combined with demand restrictions, the set of valid allocations is not a matroid but rather an intersection of two matroids. The locally greedy algorithm described in Section 2 is therefore not guaranteed to obtain a constant approximation, as shown in the following example for the case of two players. Suppose the set U of elements consists of two items, 1 and 2. Suppose player A has values 1 and $1+\epsilon$ for items 1 and 2, respectively (where $\epsilon > 0$ is arbitrarily small), and player B has values 1 and N for items 1 and 2, respectively (where $N > 1$ is arbitrarily large). When the demands of the two players are both 1, the locally greedy algorithm might allocate to either player first, but if it allocates to player A first then it obtains approximation ratio $\frac{N+1}{2+\epsilon}$, which is not bounded by a constant. The problem here stems from the asymmetry in the valuations of the two players. We introduce a notion of player anonymity[7] that captures those circumstances in which these problems do not occur.

DEFINITION 6.1. *We say that agents are* anonymous *if their valuations are symmetric; i.e. $f_i(S_1,\ldots,S_k) = f_{\pi(i)}(S_{\pi(1)},\ldots,S_{\pi(k)})$ for all permutations π and all agents $1 \leq i \leq k$.*

[6] Note that all quantities in this geometric problem are rational numbers, which are constructed via the sequence of operations described in the proof above and therefore have polynomial bit complexity. We can therefore solve the convex hull tasks described in this operation in polynomial time.

[7] Intuitively, it seems that the concepts of anonymity and Agent Indifference should be related, but neither condition implies the other. In particular, the mechanism for $k \geq 3$ players did not require anonymity.

If players are anonymous then the social welfare satisfies $f(S_1,\ldots,S_k) = f(S_{\pi(1)},\ldots,S_{\pi(k)})$ for all permutations π. We note that the models for competitive influence spread proposed by Carnes et al. [Carnes et al. 2007] and Bharathi et al. [Bharathi et al. 2007] are based on a cascade model of influence spread and assume anonymous agents.

What we will show is that when the players are anonymous, our order-independent locally greedy algorithm from Section 2 obtains a strategyproof mechanism with a $(k+1)$-approximation to the optimal social welfare, if the given permutation over orderings of the player allocations is sampled from a truthfulness-inducing distribution over permutations (e.g. the distributions we have obtained in the case of two players). Hence, this method provides a transformation to the disjoint allocations case, if one were to obtain a distribution over permutations for the non-disjoint case.

Algorithm 3 is a simple modification to Algorithm 2, in which we explicitly enforce disjointness of the allocated elements.

Algorithm 3: The Locally Greedy algorithm with disjointness

Input: Ground set $U = \{e_1,\ldots,\ldots,e_n\}$, demands a, b for players $1,\ldots,k$, a valid permutation $\pi \in \{1,\ldots,k\}^t$ where $t = \sum_{i=1}^k b_i$
Output: An allocation $I_1,\ldots,I_k \subseteq U$ for the k players

```
for i = 1 ... b_1 + ... + b_k do
    u ← argmax_{c ∈ U − ⋃_{j≠i} I_j} {w(I_i ∪ {c}, I_{−i}) − w(I_i, I_{−i})} ;
    I_i ← I_i ∪ {u} ;
end
```

THEOREM 6.1. *For any permutation $\pi \in \{1,\ldots,k\}^t$ where $t = \sum_{i=1}^k b_i$, Algorithm 3 obtains $(k+1)$-approximation to the optimal social welfare obtainable for disjoint allocation for identical players $1,\ldots,k$.*

PROOF. Let $\mathbf{O} = (O_1,\ldots,O_k)$ be an optimal allocation. Let $(I_1,\ldots,I_k)$ be the allocation obtained by running Algorithm 3 for some permutation π. Partition $\mathbf{O}$ as follows. For each player i, set $O_i^j = O_i \cap I_j$ for all $j \neq i$, and let $O_i^0 = O_i - \bigcup_{j\neq i} I_j$. By submodularity,

$$w(O_1,\ldots,O_k) \le w(O_1^0,\ldots,O_k^0) + \sum_{i=1}^{k-1} w(O_1^{(1+i) mod k},\ldots,O_k^{(k+i) mod k}). \tag{4}$$

Due to anonymity and the fact that $O_i^j \subseteq I_j$, for all $j \in [k]$ we get

$$\sum_{i=1}^{k-1} w(O_1^{(1+i) mod k},\ldots,O_k^{(k+i) mod k}) \le (k-1)\cdot w(I_1,\ldots,I_k). \tag{5}$$

Next, we can apply the analysis performed for the original locally greedy algorithm, so as to obtain the following relation (see Appendix C for further details): $w(O_1^0,\ldots,O_k^0) \le 2\cdot w(I_1,\ldots,I_k)$. Combining the two relations we get:

$$w(O_1,\ldots,O_k) \le w(I_1,\ldots,I_k) + (k-1)w(I_1,\ldots,I_k) = (k+1)\cdot w(I_1,\ldots,I_k) \tag{6}$$

□

Observe that this revised version of the locally greedy algorithm is order-independent. That is, we obtain the same (constant) bound on its approximation ratio for any player ordering. In particular, this means that we can apply the mechanism described in Section 5 for obtaining a strategyproof solution without significantly degrading the approximation ratio of the greedy algorithm.

We note that there is a natural greedy algorithm for this problem (with disjointness) that obtains a 3-approximation for any k. Namely, the greedy algorithm that chooses both the player and the allocation that maximizes the marginal utility on each iteration [Nemhauser et al. 1978a]). However, this algorithm imposes a particular ordering on the allocations and therefore does not allow the degree of freedom required by our mechanism construction.

7. CONCLUSIONS

We have presented a general framework for mechanisms that allocate items given an underlying submodular process. Although we have explicitly referred to spread processes over social networks, we only require oracle access to the outcome values, and thus our methods apply to any similar settings which uphold the properties we have required from the processes. We build on natural greedy algorithms to construct efficient strategyproof mechanisms that guarantee constant approximations to the social welfare.

An important question is how to extend our results to the more general case of $k > 2$ agents without the MeI and AgI assumptions. It seems that a fundamentally new approach would be required to obtain an $O(1)$-approximate strategyproof mechanism for $k > 2$ players. Another natural and challenging extension would be to assume that nodes have costs for being initially allocated and then replace the cardinality constraint on each agent by a knapsack constraint. To do so, the most direct approach would be to try to utilize the known approximation for maximizing a non decreasing submodular function subject to one [Sviridenko 2004] or multiple [Kulik et al. 2009] knapsack constraints. These methods do not seem to readily lend themselves to the approach we have been able to exploit in the case of cardinality constraints. We have also assumed a “demand satisfaction” condition. Without this condition, it is trivial to achieve a strategyproof k approximation by allocating all initial elements to the agent who can achieve the most utility. We would like to extend all of our results to a weaker version of “demand satisfaction” which would only require that the demand of every agent is “almost” satisfied.

8. ACKNOWLEDGEMENTS

We would like to thank Yuval Filmus for many helpful comments and discussions.

REFERENCES

Alon, N., Feldman, M., Procaccia, A. D., and Tennenholtz, M. 2010. A note on competitive diffusion through social networks. *Inf. Process. Lett. 110,* 6, 221–225.

Apt, K. and Markakis, E. 2011. Diffusion in social networks with competing products. In *Algorithmic Game Theory*, G. Persiano, Ed. Lecture Notes in Computer Science Series, vol. 6982. Springer Berlin / Heidelberg, 212–223.

Ashlagi, I., Fischer, F., Kash, I., and Procaccia, A. D. 2010. Mix and match. In *Proceedings of the 11th ACM conference on Electronic commerce*. EC ’10. ACM, New York, NY, USA, 305–314.

Babaioff, M. and Blumrosen, L. 2004. Computationally-feasible truthful auctions for convex bundles. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, K. Jansen, S. Khanna, J. Rolim, and D. Ron, Eds. Lecture Notes in Computer Science Series, vol. 3122. Springer Berlin / Heidelberg, 27–38.

Bharathi, S., Kempe, D., and Salek, M. 2007. Competitive influence maximization in social networks. In *Internet and Network Economics*, X. Deng and F. Graham, Eds. Lecture Notes in Computer Science Series, vol. 4858. Springer Berlin / Heidelberg, 306–311.

Borodin, A., Filmus, Y., and Oren, J. 2010. Threshold models for competitive influence in social networks. In *Internet and Network Economics*, A. Saberi, Ed. Lecture Notes in Computer Science Series, vol. 6484. Springer Berlin / Heidelberg, 539–550.

Carnes, T., Nagarajan, C., Wild, S. M., and van Zuylen, A. 2007. Maximizing influence in a competitive social network: a follower's perspective. In *Proceedings of the ninth international conference on Electronic commerce*. ICEC '07. ACM, New York, NY, USA, 351–360.

Dubey, P., Garg, R., and De Meyer, B. 2006. Competing for customers in a social network: The quasi-linear case. In *Internet and Network Economics*, P. Spirakis, M. Mavronicolas, and S. Kontogiannis, Eds. Lecture Notes in Computer Science Series, vol. 4286. Springer Berlin / Heidelberg, 162–173.

Goundan, P. and Schulz, A. 2007. Revisiting the greedy approach to submodular set function maximization. *Optimization online* 1984, 1–25.

Kempe, D., Kleinberg, J., and Tardos, E. 2003. Maximizing the spread of influence through a social network. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, New York, NY, USA, 137–146.

Kempe, D., Kleinberg, J., and Tardos, É. 2005. Influential nodes in a diffusion model for social networks. In *Automata, Languages and Programming*, L. Caires, G. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, Eds. Lecture Notes in Computer Science Series, vol. 3580. Springer Berlin / Heidelberg, 1127–1138.

Kostka, J., Oswald, Y. A., and Wattenhofer, R. 2008. Word of mouth: Rumor dissemination in social networks. In *Proceedings of the 15th international colloquium on Structural Information and Communication Complexity*. SIROCCO '08. Springer-Verlag, Berlin, Heidelberg, 185–196.

Kulik, A., Shachnai, H., and Tamir, T. 2009. Maximizing submodular set functions subject to multiple linear constraints. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '09. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 545–554.

Lu, P., Sun, X., Wang, Y., and Zhu, Z. A. 2010. Asymptotically optimal strategy-proof mechanisms for two-facility games. In *Proceedings of the 11th ACM conference on Electronic commerce*. EC '10. ACM, New York, NY, USA, 315–324.

Mossel, E. and Roch, S. 2010. Submodularity of influence in social networks: From local to global. *SIAM J. Comput. 39,* 6, 2176–2188.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. 1978a. An analysis of approximations for maximizing submodular set functions—II. *Mathematical Programming 14,* 1, 265–294.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. 1978b. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming 14*, 265–294. 10.1007/BF01588971.

Rockafellar, R. T. 1996. *Convex Analysis (Princeton Landmarks in Mathematics and Physics)*. Princeton University Press.

Sviridenko, M. 2004. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters 32,* 1, 41 – 43.

A. ALLOCATION OF COMMUNICATION ZONES

We now consider settings in which the agents are competing for geographical regions, which may possibly overlap. For instance, a cellphone provider would prefer to lease transmission towers which cover large areas. Consider therefore, the following setting. Given a set of communication zones, defined by disks (or any geometrical shapes located on the plane for that matter) $D_1, \ldots, D_m$, and the players' bids, we are interested in a strategyproof mechanism for allocating the players sets of disks, according to their bids, so as to maximize the total area covered. We assume that once an area has been covered, the total welfare would not change as subsequent overlapping disks are selected. Instead, the value of an area covered by several overlapping disks, will be divided by the covering disks. We first assume that the agents' valuations of each disk are identical.

This setting can be viewed as a natural extension of the model studied by Babaioff and Blumrosen in [Babaioff and Blumrosen 2004], in which overlaps are prohibited. In our case, overlaps are allowed, introducing the issue of externalities.

It is easy to see that this model follows our framework. In fact, one can verify that this problem of maximizing the covered area given the total budget can be viewed as a simple non-decreasing, submodular set-function maximization problem under a uniform matroid constraint, as the total covered area depends only on the total set of disks. Hence, we can employ the following two-phase approach. First, the algorithm will select the set of disks to be allocated, by greedily selecting at each iteration the disk which obtains the maximal marginal gain in the total area covered, until the collective budget is exhausted. Next, the mechanism will partition the set of disks among the players, according to their bids.

As an exercise, consider the simpler case in which the values overlapping areas are divided evenly among their associated disks. For instance, if overlap area of value α is covered by disks D_i and D_j, then the payoff for both disks, resulting just from that overlap, will be $\frac{\alpha}{2}$. It can be shown that for the same deterministic orderings considered for the OR model, the resulting mechanism is not strategyproof. However, for this case of unweighted agents, it can be easily shown that a uniform distribution over all possible orderings would yield a strategyproof mechanism. This follows from the fact that the expected payoff of each player will be proportional to his bid's fraction of the total budget. We also note that this particular example also satisfies the MeI and AgI assumptions.

Now, consider a slight generalization of the above scenario. For each player i, there is an associated weight w_i. For a given a region of value α, assume without loss of generality that it is covered by disks $D_1, \ldots, D_k$ allocated to players $P_1, \ldots, P_k$ (notice that some of the disks may be allocated to the same player), the fraction of value obtained by disk D_{ij} for that overlap will be proportional to the weight of player P_{i_j}, i.e. the value obtained by D_i for the overlap region is $\alpha \cdot \frac{w_i}{\sum_{i=1,\ldots,k} w_i}$. Clearly, the uniform randomized approach suggested for the unweighted case is not strategyproof. However, since the total value covered by the selected set of disks does not depend on the way it is partitioned, it does fall under the framework described in section 6.

B. COUNTEREXAMPLES WITH UNWEIGHTED NODES

In Section 4 we constructed specific examples of influence spread instances for the OR model, to illustrate that simple greedy methods are not necessarily strategyproof for the case of two players. These examples used weighted nodes; we now show that they can be extended to the unweighted node case.

We focus on the example from Section 4.3 to illustrate the idea; the other examples can be extended in a similar fashion. In that example there were nodes u_1 and u_2 of weight 1, and nodes $c_1, \ldots, c_5$ of weight 0. We modify the example as follows. We choose a sufficiently large integer $N > 1$ and a sufficiently small $\epsilon > 0$. We will replace node u_1 with a set S of $2/\epsilon$ independent nodes. We replace the ϵ-weighted edge from c_1 to u_1 with an ϵ-weighted edge from c_1 to each node in T.

Similarly, we replace u_2 by a set T of N independent nodes. For each c_i, we replace the unit-weight edge from c_i to u_2 with a unit weight edge from c_i to each node in T.

In this example, if the sum of agent budgets is at most 5, the greedy algorithm will never allocate any nodes in S or T. The allocation and analysis then proceeds just as in Section 4.3, to demonstrate that if agent B declares 1 then agent A would rather declare 3 than 4.

C. PROVING THE APPROXIMATION RATIO OF THE MECHANISM FOR THE DISJOINT CASE

Recall that in Section 6, we considered the optimal disjoint allocation $(O_1, \dots, O_k)$ to the k agents. The claim argued that if one takes the allocation $(O_1^0, \dots, O_k^k)$, such that $O_i^0 = O_i \setminus I_j$ for $i \neq j$, the following relation holds:

$$w(O_1^0, \dots, O_k^0) \leq 2 \cdot w(I_1, \dots, I_k). \tag{7}$$

We now adapt the analysis of the locally greedy algorithm (e.g. [Goundan and Schulz 2007]) in order to prove (7). We begin by introducing some additional notation. First, for $i \in [k]$, let $O_i' = O_i^0 \setminus I_i$. For an item $e \in I_i$ ($i \in [k]$), $\mathbf{S}^e = (S_1^e, \dots, S_k^e)$ denotes the partial solution of the algorithm at the time of item e's addition. For an allocation $(A_1, \dots, A_k) \subseteq U^k$ and an item $e \in U$, define the marginal gain to be $\rho_e^i(A_1, \dots, A_k) = w(A_1, \dots, A_i \cup \{e\}, \dots, A_k) - w(A_1, \dots, \dots, A_k)$. Lastly, for $i \in [k]$, we let $e_i = \arg\min_{e \in I_i} \rho_e^i(\mathbf{S}^e)$; i.e. the minimal marginal increase to social welfare, obtained by the algorithm, when adding an element to I_i. The following lemma follows from the fact that the social welfare is a non-decreasing, submodular function:

LEMMA C.1 ([NEMHAUSER ET AL. 1978B]).

$$w(O_1^0, \dots, O_k) \leq w(I_1, \dots, I_k) + \sum_{i=1}^{k} \sum_{e \in O_i'} \rho_e^i(I_1, \dots, I_k)$$

Now, by the greedy rule of Algorithm 3, we have:

$$\rho_{e_i}^i(S_i^e) \geq \rho_e^i(S_i^e), \text{for all } i \in [k], \text{ and for all } e \in U \setminus S_i^{e_i} \tag{8}$$

Additionally, using the submodularity of $w(\cdot)$, and the fact that for all $i, j \in [k]$, $S_j^{e_i} \subseteq I_j$, we get:

$$\begin{aligned} w(O_1^0, \dots, O_k^0) &\leq w(I_1, \dots, I_k) + \sum_{i=1}^{k} \sum_{e \in O_i'} \rho_e^i(I_1, \dots, I_k) \\ &\leq w(I_1, \dots, I_k) + \sum_{i=1}^{k} \sum_{e \in O_i'} \rho_e^i(\mathbf{S}^{e_i}) \end{aligned} \tag{9}$$

Now, using (8), we further extend the above bound as follows

$$\begin{aligned} w(O_1^0, \dots, O_k^0) &\leq w(I_1, \dots, I_k) + \sum_{i=1}^{k} \sum_{e \in O_i'} \rho_{e_i}^i(\mathbf{S}^{\mathbf{e_i}}) \\ &= w(I_1, \dots, I_k) + \sum_{i=1} b_i \cdot \rho_{e_i}^i(\mathbf{S}^{\mathbf{e_i}}) \\ &\leq w(I_1, \dots, I_k) + w(I_1, \dots, I_k) = 2 \cdot w(I_1, \dots, I_k) \end{aligned} \tag{10}$$

where the last inequality follows from the definition of the elements $e_1, \dots, e_k$.

D. TIGHTNESS OF APPROACH: MORE THAN TWO PLAYERS

The mechanism we construct in Section 5 is applicable to settings in which there are precisely two competing players, and our mechanism in Section 3 for more than three players requires the MeI and AgI assumptions. A natural open question is whether these results can be extended to the general case of three or more agents without the MeI and AgI restrictions. In this section we briefly describe the difficulty in applying our approach to settings with three players.

For the case of two players in Section 5, our mechanism was built from an initial greedy algorithm by randomizing over orderings under which to assign elements to players. Our construction is recursive: we demonstrated that if we can define the behaviour of a strategyproof mechanism for all possible budget declarations up to a total of at most k, then we can extend this to a strategyproof mechanism for all possible budget declarations that total at most $k+1$. One key observation that makes this extension possible is that the strategyproofness condition can be re-expressed as a certain "adverse competition" property: if one player increases his budget, then the expected utility for the other player cannot increase by more than the marginal gain in total welfare. In other words, in the notation of Lemma 5.1, we can construct our mechanism so that for all $a+b \geq 1$, $w^A(a,b) \leq w^A(a,b-1) + \Delta^{\oplus B}(a,b)$.

A direct extension of our approach to three players would involve proving that an allocation rule that is strategyproof and satisfies the adverse competition condition for all budgets that total at most k can always be extended to handle budgets that total up to $k+1$. We now give an example to show that this is not the case, even when our underlying submodular function takes a very simple linear form.

Suppose we have three players A, B, and C, and suppose our ground set U contains a single element c of value; all other elements are worth nothing. The utility for each agent is 1 if their allocation contains c, otherwise their utility is 0. In this case, the locally greedy algorithm simply gives element c to the first player that is chosen for allocation; the remaining allocations have no effect on the utility of any player. Note then that the marginal gain in social welfare is 1 for the first allocation, and 0 for all subsequent allocations made by the greedy algorithm.

We now define the behaviour of a mechanism for all budget declarations totalling at most 2. Note that the relevant feature of this mechanism is the (possibly randomized) choice of which agent is first in the order presented to the greedy algorithm. We present this behaviour in the following table.

Budgets (a,b,c)	Player selected	Utilities (w^A, w^B, w^C)
$(0,0,0)$	N/A	$(0,0,0)$
$(1,0,0)$	A	$(1,0,0)$
$(0,1,0)$	B	$(0,1,0)$
$(0,0,1)$	C	$(0,0,1)$
$(1,1,0)$	A	$(1,0,0)$
$(0,1,1)$	B	$(0,1,0)$
$(1,0,1)$	C	$(0,0,1)$

We note that this mechanism (restricted to these type profiles) is strategyproof, satisfies the adverse competition property, and also satisfies the cross-monotonicity property (i.e. the first invariant of Lemma 5.1). However, we claim that no allocation on input $(1,1,1)$ that obtains positive social welfare can maintain the adverse competition property. To see this, note that the adverse competition property would imply that $w^A(1,1,1) \leq w^A(1,0,1) + \Delta^{\oplus B}(1,1,1) = w^A(1,0,1) = 0$. Similarly, we must have $w^B(1,1,1) = w^C(1,1,1) = 0$. Thus, in order to maintain the adverse competition property, our mechanism would have to generate social welfare 0 on input $(1,1,1)$, resulting in an unbounded approximation factor. We conclude that there is no way to extend this mechanism for budgets totalling at most 2 to a (strategyproof, cross-monotone) mechanism for budgets totalling at most 3 while maintaining the constant approximation factor of the locally greedy algorithm.

Roughly speaking, the problem illustrated by this example is that the presence of more than two bidders means that a substantial increase in the utility gained by one player does not necessarily imply a decrease in the utility of each other player. This is in contrast to the case of two players, in which the utilities of the two players are more directly related.

This fundamental difference seems to indicate that substantially different techniques will be required in order to construct strategyproof mechanisms with three or more players.

A different (and natural) approach would be to employ the solution for two players by grouping all but one player at a time, and running the mechanism for two players recursively. However, this method seems ineffective in our setting, as interdependencies between the players' outcomes can introduce non-monotonicities. Hence, we believe that our greedy mechanism cannot be made strategyproof via our general method of randomizing over the order in which allocations are made.

This "2 vs 3 barrier" is, of course, not unique to our problem. Many optimization problems (such as graph coloring) are easily solvable when the size parameter is $k = 2$ but become NP-hard when $k \geq 3$. Closer to our setting, the 2 vs 3 barrier has been discussed in recent papers concerning mechanism design without payments, such as in the Lu et al. [Lu et al. 2010] results for k-facility location. Additionally Ashlagi et al. discussed similar issues ([Ashlagi et al. 2010]) in the context of mechanisms for kidney exchange. They show that for n points on the line, there is a deterministic (respectively, randomized) strategyproof mechanism for placing $k = 2$ facilities (so as to minimize the sum of distances to the nearest facility) with approximation ratio $n - 2$ (respectively, 4) whereas for $k = 3$ facilities, they do not know if there is any bounded ratio for deterministic strategyproof mechanisms and the best known approximation for randomized strategyproof mechanisms is $O(n)$.