The Importance of System-Level Information in Multiagent Systems Design: Cardinality and Covering Problems

Dario Paccagnan¹ and Jason R. Marden²

Abstract

A fundamental challenge in multiagent systems is to design local control algorithms to ensure a desirable collective behaviour. The information available to the agents, gathered either through communication or sensing, naturally restricts the achievable performance. Hence, it is fundamental to identify what piece of information is valuable and can be exploited to design control laws with enhanced performance guarantees. This paper studies the case when such information is uncertain or inaccessible for a class of submodular resource allocation problems termed covering problems. In the first part of this work we pinpoint a fundamental risk-reward tradeoff faced by the system operator when conditioning the control design on a valuable but uncertain piece of information, which we refer to as the cardinality, that represents the maximum number of agents that can simultaneously select any given resource. Building on this analysis, we propose a distributed algorithm that allows agents to learn the cardinality while adjusting their behaviour over time. This algorithm is proved to perform on par or better to the optimal design obtained when the exact cardinality is known a priori.

I. INTRODUCTION

SEVERAL social and engineering systems can be thought of as a collection of multiple subsystems or agents, each taking local decisions in response to available information. A central goal in this field is to design control algorithms for the individual subsystems to ensure that the collective behaviour is desirable with respect to a global objective. Achieving this goal is particularly challenging because of the restriction on the information available to each agent and to the large scale of typical systems. Examples include, but are not limited to, power grid networks [2], charging of electric vehicles [3], transportation network [4], task assignment problems [5], sensor allocation [6], robotic networks [7]. A considerable bulk of the research has focused on

This research was supported by SNSF Grant #P1EZP2-172122 and by AFOSR Grant #FA9550-12-1-0359, ONR Grant N00014-15-1-2762, NSF Grant #ECCS-1351866.

1D. Paccagnan is with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, USA and with the Automatic Control Laboratory, Swiss Federal Institute of Technology, Zurich, Switzerland. Email: dariop@control.ee.ethz.ch.

2J. R. Marden is with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, USA. Email: jrmarden@ece.ucsb.edu.

A much abbreviated conference version of this work appeared in [1].
the design of local control algorithms in a framework where the information at agents’ disposal is itself a fixed datum of the problem. A non exhaustive list includes [8], [9] and references therein. Understanding the impact of information availability on the achievable performances is a seemingly important but less tracked problem [10], [11], [12].

Of particular interest is to recognise what supplementary piece of information could coordinate the agents to improve the system performance, and further how to incorporate this additional knowledge into a control algorithm. It is important to highlight that providing each agent with all the information available to the system is in principle beneficial, but not necessarily desirable. Indeed, the communication costs associated with propagating additional information through the system might overcome the performance gains that the knowledge of additional information gives. Therefore, the previous question has to be understood within this context. Ideally, one is interested in a piece of information that gives a significant performance enhancement, and is simple to obtain. Loosely speaking, we measure the value of an additional piece of information with the performance gain that the best controller can offer, using that supplementary piece of information.

Relative to the class of resource allocation problems termed covering problems, [11], [13] show that the maximum number of agents that can simultaneously select a resource (which we term cardinality, see Equation (1) for a formal definition) constitutes a valuable piece of information. More precisely, when the system operator is aware of the cardinality of the problem, he can devise distributed algorithms with improved performance guarantees. Nevertheless, the knowledge of the exact cardinality is in many applications uncertain, not available or may require excessive communication to be determined. Following this observation, a system operator would like to understand how to operate when the knowledge of the exact cardinality is not available. What is the risk associated with using the wrong cardinality in the control design? What is the reward for using the correct one? Further and more fundamental: when the cardinality is not available at all, can the agents learn it while simultaneously adjusting their behaviour?

The paper proceeds by considering covering problems [14], [15], a class of resource allocation problems where agents are assigned to resources in order to maximise the total value of covered items. Examples include vehicle-target assignment problems [16], sensor allocation [6], task assignment [17], among others. Due to the inherent limitations in sensing and communication, in all these applications the control algorithms are required to rely only on local information. Thus, we model distributed covering problems as strategic-form games, where the system operator has the ability to assign local objective functions to each agent. Indeed, as shown in [18], [10], Game Theory lends itself to analyse distributed systems where individual agents adjust their behaviour in response to partial information. Such game theoretic approach offers the possibility to build upon existing tools to quantify the system performance, as well as the opportunity to exploit readily available algorithms to compute equilibria in a distributed fashion [5], [13]. The
overarching goal of the system operator is to design local utilities in order to render the equilibria of the game as efficient as possible. Agents can then be guided towards an equilibrium of such game by means of existing distributed algorithms [5], [19]. It is important to highlight that we are not modelling agents as competing units, but we are rather designing their utilities to achieve the global objective.

Building on the previous results of [11], [13], we contribute as follows.

i) We study the problem of optimally designing the utility functions in the case when the true cardinality is not known, but only an upper bound is available. We further perform a risk-reward analysis in the case when the information on the cardinality of the game is uncertain. When the goal is to guard the system against the worst case performances, the right choice is to design the utilities as if the true cardinality was the given upper bound. Different designs will offer potential benefits, but come with a certain degree of risk. These results are presented in Theorem 1.

ii) Motivated by the potential advantages and inherent shortcomings presented in the risk-reward analysis, we propose a distributed and asynchronous algorithm that dynamically updates the utility functions while agents adjust their behaviour over time. Such algorithm requires no initial information, and is certified to perform on par or better (in a worst case sense) to the optimal design possible, had we known the cardinality in the first place. These results are summarised in Theorem 2.

iii) We compare, instance by instance, the performance of the proposed learning algorithm with the performance of the optimal design obtained with full knowledge of the cardinality. We show that it is not possible to deem one approach superior to the other on all instances of covering problems, in that there are instances where one outperforms the other, and the reverse too. These results are presented in Theorem 3.

The remaining of the paper is organised as follows. The next section introduces the covering problem, its formulation as a strategic game and the metric used to measure the system-level performance. Section III studies the utility design problem when a sole upper bound on the cardinality is available and presents the risk-reward tradeoff associated with the use of uncertain information. Section IV shows the possibility of dynamically adjusting the utility functions to improve the performance. Numerical simulations and conclusions follow.

Notation

For any two positive integers \( p \leq q \), denote \([p] = \{1, \ldots, p\}\) and \([p, q] = \{p, \ldots, q\}\); given \((a^1, \ldots, a^n)\), denote \(a^{-i} = (a^1, \ldots, a^{i-1}, a^{i+1}, \ldots, a^n)\). We use \( \mathbb{N} \) and \( \mathbb{R}_{\geq 0} \) to denote the set of natural numbers (excluding zero) and the set of non-negative real numbers, respectively.

1A simple bound is given by the number of agents.
II. DISTRIBUTED COVERING VIA GAME THEORY

In this section we present the covering problem and the associated covering game. We further define the performance metric used throughout the paper and recap previous results.

A. Model

Let us consider the problem of assigning a collection of agents $N = \{1, \ldots, n\}$ to a finite set of resources $\mathcal{R} = \{r_1, \ldots, r_m\}$ with the goal of maximising the value of covered resources. The feasible allocations for each agent $i \in N$ are the elements of the action set $a^i \in \mathcal{A}^i \subseteq 2^\mathcal{R}$, while every resource $r \in \mathcal{R}$ is associated with a non-negative value $v_r \geq 0$. Observe that $a^i \subseteq \mathcal{R}$. The welfare of an allocation $a = (a^1, \ldots, a^n) \in \mathcal{A}^1 \times \cdots \times \mathcal{A}^n$ is measured by the total value of covered resources

$$W(a) := \sum_{r: |a|_r \geq 1} v_r,$$

where $|a|_r$ denotes the number of agents that choose resource $r$ in allocation $a$. The covering problem $C = \{N, \mathcal{R}, \{\mathcal{A}^i\}_{i \in N}, \{v_r\}_{r \in \mathcal{R}}\}$ consists in finding an optimal allocation, that is an assignment $a_o \in \arg \max_{a \in \mathcal{A}} W(a)$.\footnote{Approximation algorithms for finding a near optimal solution to submodular optimization problems have been extensively studied in the literature [20], [21]. The focus of this literature is predominantly on centralized algorithms for finding near optimal allocations. In contrast, our focus is on distributed solutions where each decision-making entity has incomplete information about the system as a whole.} Given a covering problem $C$, we define its cardinality as the maximum number of players that can concurrently select the same resource, that is

$$\max_{r \in \mathcal{R}, a \in \mathcal{A}} |a|_r. \quad (1)$$

Instead of directly specifying a distributed algorithm, we shift the focus to the design of local utility functions for each agent, as proposed first for distributed welfare games by [22], [5] and successively by [11]. Within this framework, each agent $i \in N$ is associated with a utility function of the form

$$u^i(a^i, a^{-i}) := \sum_{r \in \mathcal{A}^i} v_r \cdot f(|a|_r). \quad (2)$$

The function $f: [n] \rightarrow \mathbb{R}_{\geq 0}$ constitutes our design choice and is called distribution rule as it represents the fractional benefit an agent receives from each resource he selects. The advantages of using utilities of the form (2) are twofold. First, $u^i(a^i, a^{-i})$ is local as it depends only on the resources agent $i$ selects, their value and the number of agents that selects the same resources. Second, (2) allows to construct a distribution rule irrespective of $\{\mathcal{A}^i\}_{i \in N}$ and $\{v_r\}_{r \in \mathcal{R}}$ so that the final design is scalable and applies to different choices of the action sets and of the resource valuations.
Given a covering problem $C$ and a distribution rule $f : [n] \to \mathbb{R}_{\geq 0}$, we consider the associated covering game $G := \{C, f\} = \{N, \mathcal{R}, \{A^i\}_{i \in N}, \{v_r\}_{r \in \mathbb{R}}, f\}$, where $A^i$ is the set of feasible allocations and the utility of agent $i \in N$ is as in equation (2).

We do not aim at designing $f$ using information on the specific instance of covering problem at hand, as such information is often not available to the system designer. Our goal is rather to construct a distribution rule that behaves well for a large class of problems. Hence, we consider the set of covering problems for which the cardinality is smaller or equal to $k \in \mathbb{N}, k \leq n$.

Given a distribution rule $f : [k] \to \mathbb{R}_{\geq 0}$, we define the set of associated games as

$$G^k_f := \{G = \{C, f\} : \max_{r \in \mathbb{R}, a \in A} |a|_r \leq k\}.$$

Our objective is to design $f : [k] \to \mathbb{R}_{\geq 0}$ so that the efficiency of all the equilibria of games in $G^k_f$ is as high as possible. Note that for fixed $f$, any game $G$ is potential [22]. Hence existence of equilibria is guaranteed and distributed algorithms, such as the best response scheme, converge to them [23]. Throughout the paper, we focus on pure Nash equilibria [24], which we will refer to in the following just as equilibria.

**Definition 1** (Pure Nash equilibrium). Given a game $G$, an allocation $a_e \in A$ is a pure Nash equilibrium iff $u^i(a^i_e, a^{-i}_e) \geq u^i(a^i, a^{-i}_e)$ for all deviations $a^i \in A^i$ and for all players $i \in N$. In the following we use $\text{NE}(G)$ to denote the set of Nash equilibria of $G$.

For a given distribution rule, we evaluate the efficiency of the Nash equilibria of games in $G^k_f$, adapting the concept of Price of Anarchy from [25] as

$$\text{PoA}(f, k) := \inf_{G \in G^k_f} \left\{ \frac{\min_{a \in \text{NE}(G)} W(a)}{\max_{a \in A} W(a)} \right\} \leq 1.$$  \hspace{1cm} (3)

In essence, the quantity $\text{PoA}(f, k)$ bounds the inefficiency of the worst equilibrium (and thus of all equilibria) over games in $G^k_f$, that is over games with distribution rule set to $f$ and cardinality bounded by $k$.\footnote{The quantity $W(a)$ appearing in Equation (3) does depend on which game instance $G$ we are considering, since the resource valuations do. Hence, a more formal notation would entail using $W(a; G)$. In the interest of readability, we avoid the latter and simply use $W(a)$ when no ambiguity arise.} The higher the price of anarchy, the better the performance guarantees we can provide.

**B. Related Work and Performance Guarantees**

The problem of designing a distribution rule so as to maximise $\text{PoA}(f, k)$ has been studied in [11] and [13]. Both works impose a natural constraint on the admissible $f$, requiring $f(1) = 1$ and $f : [k] \to \mathbb{R}_{\geq 0}$ to be non-increasing. The optimal distribution rule is explicitly derived in...
the former work, while the latter shows how $\text{PoA}(f, k)$ is fully characterised by a single scalar quantity $\chi(f, k)$ defined in (4), measuring how fast the distribution rule $f$ decreases. We intend to build upon these results, which are summarised in the following proposition. Given $k$ and a distribution rule $f$, we define

$$\chi(f, k) := \max_{j \leq k-1} \{ j \cdot f(j) - f(j+1), (k-1) \cdot f(k) \}.$$  \hfill (4)

**Proposition 1** ([11], [13]). Consider a non-increasing distribution rule $f : [k] \to \mathbb{R}_{\geq 0}$, with $f(1) = 1$.

i) The price of anarchy over the class $G^k_f$ is

$$\text{PoA}(f, k) = \frac{1}{1 + \chi(f, k)}.$$  

ii) The price of anarchy over the class $G^k_f$ is maximised for

$$f^*_k(j) = (j - 1) \frac{1}{(k-1)(k-1)!} + \sum_{i=j}^{k-1} \frac{1}{i!}, \quad j \in [k]$$  \hfill (5)

with corresponding

$$\chi(f^*_k, k) = (k-1) \cdot f^*_k(k).$$  \hfill (6)

iii) The optimal price of anarchy is a decreasing function of the cardinality $k$

$$\text{PoA}(f^*_k, k) = 1 - \frac{1}{(k-1)(k-1)!} + \sum_{i=1}^{k-1} \frac{1}{i!}.$$  \hfill (7)

**III. The Case of Unknown Cardinality: A Risk-Reward Tradeoff**

When the cardinality $k$ defining the class of games $G^k_f$ is known, Proposition 1 gives a conclusive answer on which distribution rule agents should choose to achieve the best worst case performance. In spite of that, the knowledge of the exact cardinality is in many applications not available or may require excessive communications between the agents to be determined.

Motivated by this observation, we study in the following the problem of designing a distribution rule when the cardinality $k$ defining the class of games $G^k_f$ is not known, but an upper bound $k \leq k_u$ is available. Observe that a universal upper bound for such quantity can be easily computed as the number $n$ of agents. Potentially tighter bounds can be derived for specific applications. Our objective is to design a distribution rule $f : [k_u] \to \mathbb{R}_{\geq 0}$ with the best performance guarantees possible with the sole knowledge of $k_u$. Once such a distribution rule has been designed, one can use existing distributed algorithms to find an equilibrium as discussed in the introduction. Two natural questions arise in this context:

1) How should we select the distribution rule?

2) What performance can we guarantee?
We will show how selecting \( f^*_{ku} \) guards us against the worst case performance but will not guarantee the same efficiency of \( f^*_k \), when \( k < k_u \). We will then present the potential benefits and risks associated with a more aggressive choice. These results motivate Section IV, where we will present a dynamic scheme that overcomes the difficulties encountered here, offering the same performances of \( f^*_k \) at no risk.

A. Two alternative distributions

A natural choice when an upper bound on the cardinality is available consists in designing the distribution rule exactly at the upper bound, that is using \( f^*_{ku} \). A different choice might entail constructing a distribution rule where the entries \([k_d]\) are designed as if the cardinality was \( k_d < k_u \), while the remaining entries \([k_d + 1, k_u]\) are optimally filled. The latter suggestion is inspired by the observation that the optimal system level performance (measured by the price of anarchy) is a decreasing function of \( k \) as per (7). This distribution is denoted with \( f'_{kd} \) and is constructed fixing \( f'_{kd}(j) = f^*_{kd}(j) \) for \( j \in [k_d] \). The tail entries corresponding to \( j \in [k_d + 1, k_u] \) are chosen to mitigate the risk taken. Formally, for any \( 1 < k_d < k_u \) we define the distribution rule \( f'_{kd} : [k_u] \to \mathbb{R}_{\geq 0} \) as a solution of the following optimisation problem

\[
\begin{align*}
\text{s.t. } f(j) &= f^*_{kd}(j) \quad \forall j \in [k_d],
\end{align*}
\]

where \( F = \{ f : [k_u] \to \mathbb{R}_{\geq 0} \text{ with } f(1) = 1, f(j + 1) \leq f(j), \forall j \in [k_u - 1] \} \) is the set of admissible distributions.

Note that we do not define \( f'_{kd} \) for \( k_d = 1 \) or \( k_d = k_u \) as it would reduce in both cases to \( f^*_{ku} \). Further observe that the constraint \( f'_{kd}(j) = f^*_{kd}(j), \forall j \in [k_d] \) is equivalent to requiring \( f'_{kd} \in \arg \max_{f \in F} \text{PoA}(f, k_d) \).

The next proposition characterises explicitly \( f'_{kd} \).

**Proposition 2.** For any \( 1 < k_d < k_u \), the distribution \( f'_{kd} \) defined in (8) is given by

\[
\begin{align*}
f'_{kd}(j) &= \begin{cases} f^*_{kd}(j) & j \in [k_d] \\ \frac{(j-1)!}{(k_d-1)!} f^*_{kd}(k_d) - \chi(f'_{kd}, k_u)(\sum_{h=1}^{j-1-k_d} \frac{(j-1)!}{(j-h-1)!} + 1) & j \in [k_d + 1, k_u] \end{cases}
\end{align*}
\]

where the expression for \( \chi(f'_{kd}, k_u) \) can be found in Equation (13) in the Appendix.

The proof can be found in the appendix.

**Remark.** In [11] a distribution rule \( f \) was required to satisfy the constraint \( j \cdot f(j) \leq 1 \) for all \( j \). Loosely speaking the above requirement guarantees that a distribution rule does not overpay
the players, in the sense that $\sum_{i \in N} u^i(a^i, a^{-i}) \leq W(a)$ for all allocations. Observe that such constraint might be important for economic applications, but it is irrelevant in the design of engineering systems. While [11] shows that the distribution rule $f^*_k$ satisfies this property, the next lemma proves that also $f'_{k_d}$ verifies this condition even if this was not requested a priori.

**Lemma 1.** For any $1 < k_d < k_u$ the distribution $f'_{k_d} : [k_u] \rightarrow \mathbb{R}_{\geq 0}$ satisfies $j \cdot f'_{k_d}(j) \leq 1$ for all $j \in [k_u]$.

The proof is provided in the appendix.

**B. Performance comparison**

Based on the metric introduced in (3), we compare in this section the performance of $f^*_k$ with the performance of $f'_{k_d}$. Theorem 1 constitutes the main result of this section.

**Theorem 1.** Consider the set of games $G^k_f$, where $k \leq k_u$.

i) The distribution $f^*_k$ has performance

$$\text{PoA}(f^*_k, k) = \text{PoA}(f^*_k, k_u).$$

Such performance is strictly worse than the one achieved by the distribution $f^*_k$ if $k < k_u$ and equal if $k = k_u$.

ii) For $1 < k_d < k$ the distribution $f'_{k_d}$ has performance

$$\text{PoA}(f'_{k_d}, k) = \text{PoA}(f'_{k_d}, k_u),$$
which is strictly worse than the one achieved by $f^*_{k_u}$.

For $k \leq k_d < k_u$ the distribution $f'_{k_d}$ has performance

$$\text{PoA}(f'_{k_d}, k) = \text{PoA}(f^*_{k_d}, k_d),$$

which is strictly better than the one achieved by $f^*_{k_u}$.

The proof can be found in the Appendix.

Remark. Claim i) in Theorem 1 shows that the performance of the distribution $f^*_{k_u}$ on the class of games with cardinality bounded by $k$ is independent on the actual value of $k$, as for any $k \leq k_u$, such performance is governed by $\text{PoA}(f^*_{k_u}, k_u)$. Claim ii) in Theorem 1 ensures that the distribution $f'_{k_d}$ outperforms $f^*_{k_u}$ for $k_u > k_d \geq k$ and the opposite when $k_d < k$. In each of these cases the performance is independent on the actual value of $k$, but only depends on whether $k_d$ is above or below $k$. Loosely speaking, if we underestimate $k$ by designing $k_d < k$, the performance guarantees offered by $f'_{k_d}$ are worse than what $f^*_{k_u}$ can achieve. The reverse holds in the case when we overestimate the cardinality as in $k_u > k_d \geq k$.

In Figure 2 we compare the performance of $f'_{k_d}$ with the performance of $f^*_{k_u}$. It is important to note that the performance degradation (incurred whenever $k_d < k$) always dominates the potential gains (achieved when $k_d \geq k$). This motivates the next section where we will introduce a dynamic algorithm capable of offering the benefits of $f'_{k_d}$ for $k_d \geq k$ at no risk.

IV. BEYOND THE RISK-REWARD TRADEOFF

The previous section has focused on the design of a distribution rule when an upper bound on the true cardinality is known. We have demonstrated how $f^*_{k_u}$ guards against worst case performance while $f'_{k_d}$ could give potential benefits, but comes with a certain degree of risk. In both cases the performance is equal or inferior to what we could achieve if we knew the true cardinality.

In this section we show how to overcome such difficulties, when we are given a game $G \in G^k_f$ with unknown cardinality $k$. We propose a distributed and asynchronous implementation of the best response algorithm that dynamically updates which distribution rule to use (see footnote 5 for further details). The upshot is that we guarantee an equal or superior performance to what we could achieve if we knew $k$.

In the following, we allow distribution rules to depend on an additional variable $x_r \in [n]$ defined for $r \in \mathcal{R}$, which we will dynamically update to coordinate the agents. In particular, we generalise the utilities of (2) to

$$u^i(a^i, a^{-i}; x) := \sum_{r \in a^i} v_r f(x_r, |a|_r), \quad (10)$$
Fig. 2: The bars represent the difference $\text{PoA}(f_{k_d}^{'}, k) - \text{PoA}(f_{k_u}^{'}, k)$, normalized by its largest value. As such, it describes the normalized difference in performance between $f_{k_d}^{'}, f_{k_u}^{*}$ for various values of $1 < k_d < k_u = 10$. In the first figure $k = 3$, while $k = 4$ in the second figure.

where $x = \{x_r\}_{r \in R}$ and $f : [n] \times [n] \rightarrow \mathbb{R}_{\geq 0}$ might be different across the resources, depending on the value of $x_r$.

One could question whether the improved performance we will obtain comes from the additional degree of freedom introduced allowing resource specific distribution rules. Nevertheless [11] shows that it is not the case, in that the best resource specific and non resource specific distribution perform equally (in the worst case sense). The only rationale to introduce resource dependent
rules is the distributability of the algorithm. Indeed, similar results could have been achieved dynamically updating a single distribution rule shared by all resources, but such algorithm would have not been distributed.

A. Algorithm description and distributedness

In the following $t \in \mathbb{N}$ describes the time step of the algorithm and $a_t \in \mathcal{A}$ the corresponding allocation. With slight abuse of notation, for every resource $r \in \mathcal{R}$ we introduce the quantity $x_r(t)$ that associates $r \in \mathcal{R}$ to the maximum number of agents that chose such resource until time $t \in \mathbb{N}$. Further, we define $f^{\text{alg}}_\ell : [n] \to \mathbb{R}_{\geq 0}$ for every $\ell \in \mathbb{N}$ as a distribution rule matching the optimal in equation (5) for $j \in [\ell]$ and constant in between $[\ell, n]$$^4$

\[
 f^{\text{alg}}_\ell(j) := \begin{cases} 
   f^*_\ell(j) & j \in [\ell], \\
   f^*_\ell(\ell) & j \in [\ell + 1, n]. 
\end{cases}
\] (11)

Algorithm 1 Asynchronous cardinality learning

1: Initialise $a_1$; \quad $t \leftarrow 1$; \quad $x_r(t) \leftarrow |a_1|_r$ \quad $\forall r \in \mathcal{R}$

2: \textbf{while} not converged \textbf{do}

\text废弃
3: \textbf{Best response}
   \quad $i \leftarrow t \mod n$
4: \quad $a_{t+1} \leftarrow \arg \max_{a_t \in \mathcal{A}_i} \sum_{r \in a_t} v_r f^{\text{alg}}_{x_r(t)}(|a_t|_r)$
5: \quad $a_{t+1} \leftarrow (a_{t+1}, a_t - i)$

\text废弃
6: \textbf{Update} $k_t$ and thus $f$ on every resource
   \quad $x_r(t + 1) \leftarrow \max\{x_r(t), |a_{t+1}|_r\}$ \quad $\forall r \in \mathcal{R}$
7: \quad $t \leftarrow t + 1$
8: \textbf{end while}

Through the additional variable $x_r(t)$, the algorithm keeps track of the maximum number of players that visited every resource until the current time $t$, and selects consequently a resource specific distribution rule. In particular on every $r \in \mathcal{R}$, the algorithm uses $f^{\text{alg}}_\ell$ with $\ell$ set to the maximum number of players that visited that resource until time $t$ (lines 4 and 6). Following a

$^4$The rule $f^{\text{alg}}_\ell$ is a valid distribution rule, being non increasing and such that $f^{\text{alg}}_\ell(1) = 1$. It will in general not satisfy $j \cdot f^{\text{alg}}_\ell(j) \leq 1$, but this was neither requested, nor has relevance in the design of engineering systems.
round-robin rotation, players \( i \) is selected to best respond and update the allocation (lines 3 to 5). The procedure repeats until convergence.\(^5\)

The algorithm is distributed in the sense that every agent needs to keep track of \( x_r(t) \) only for those resources he has access to i.e. for \( r \in A^i \). Further, it is asynchronous as players need not to update their allocation in a specified order, but can spontaneously revise their strategies. It is important to highlight that the communication requirements of Algorithm 1 are the same of those needed by the best response algorithm applied for instance to distribution rules \( f^*_{k,M} \) or \( f'_{k,M} \). That is, Algorithm 1 better exploits the information that is already available.

For ease of exposition we have presented the case where the distribution rules depend on the history \( x_r(t) \), but the same across the players. It is simple to extend these results to the case of agent specific distribution rules. Every player would use a resource specific distribution rules that depend on the maximum number of players that visited every resource up until his last visit. Similar convergence guarantees and performance certificates will follow.

**B. Convergence and quality of equilibria**

The following theorem is the main result of this section. Claim i) shows convergence of Algorithm 1 to a Nash equilibrium. Claim ii) proves that the quality of such equilibrium is higher or equal to what the optimal distribution \( f^*_{k,M} \) could achieve.

**Theorem 2.** Consider a covering game \( G \) with cardinality \( k \) and initial assignment \( a_1 \in A \).

i) Algorithm 1 converges in a finite number of steps to \( a_e := \lim_{t \to \infty} a_t \in A \). The allocation \( a_e \) is a Nash equilibrium of the game with resource specific distribution rules fixed to \( f^\text{alg}_{x_r^\infty} \) for \( r \in R \), where \( x_r^\infty := \lim_{t \to \infty} x_r(t) \).

ii) Let \( k_M := \max_{r \in R} x_r^\infty \). The quality of the equilibrium \( a_e \) is higher than \( \text{PoA}(f^*_{k,M}, k_M) \) and thus of \( \text{PoA}(f^*_{k}, k) \)

\[
\frac{W(a_e)}{W(a_o)} \geq \text{PoA}(f^*_{k,M}, k_M) \geq \text{PoA}(f^*_{k}, k). \tag{12}
\]

These statements hold for any initial condition, even if the allocation to which the Algorithm 1 converges may be different. The proof is detailed in the Appendix.

\(^5\)Note that the best response strategy is not guaranteed to be unique. To overcome this issue, in the following we assume the existence of a tie-breaking rule selecting a single optimal allocation, should these be multiple. Nevertheless, we observe that neither this, nor requiring players to best respond in a round-robin is fundamental. It is still possible to show that Algorithm 1 converges almost surely if the players best responding are uniformly randomly selected form \([n]\) and a single optimal allocation is uniformly randomly extracted from the set of best responses. This will produce a totally asynchronous algorithm.
C. Instance by instance analysis

The previous theorem shows that Algorithm 1 achieves a higher or equal worst case performance than the optimal distribution $f_k^*$. While worst case analysis has been and still is a fruitful tool to measure and improve on algorithms’ performance, the computer science community has recently showed interest in moving beyond it [26]. Inspired by this, the question arises as to whether Algorithm 1 performs better than $f_k^*$, instance by instance. More formally, we would like to understand if Algorithm 1 yields higher welfare than the optimally designed rule on all the remaining instances (the non worst case ones). We show that neither this, nor the opposite holds.

**Theorem 3.** Let $C$ be an instance of covering problem defined in Section II. Further denote with $\text{NE}^{\text{alg}}(C)$ the set equilibria obtained using Algorithm 1 on $C$, and $G^* = \{C, f_k^*\}$ the associated game where the optimal distribution $f_k^*$ has been selected.

i) There exists an instance $C$ of the covering problem such that

$$\min_{a \in \text{NE}^{\text{alg}}(C)} W(a) > \min_{a \in \text{NE}(G^*)} W(a).$$

ii) There exists an instance $C$ of the covering problem such that

$$\min_{a \in \text{NE}^{\text{alg}}(C)} W(a) < \min_{a \in \text{NE}(G^*)} W(a).$$

The proof is constructive and is presented in the Appendix. Note that both statements in Theorem 3 compare the performances of a given covering problem $C$ and associated game $G^*$. Observe that this metric is significantly different from (3), where we additionally take the infimum over problems with cardinality bounded by $k$.

V. Simulations

In this section we provide simulations to compare the performance of different distribution rules. More specifically, given a coverage problem of cardinality $k$, we compare the performance of Algorithm 1 with the performance of the best response dynamics (BR) using the distribution rule $f_{k,u}$ or the optimal distribution rule $f_k^*$.

For this numerical study, we randomly generated 3000 instances of the covering problem with $n = 10$ agents, $m = 20$ resources. Each agent is equipped with an action set whose elements are singletons, that is $a_i \in \mathcal{R}$ for all $i \in N$. We believe this is not restrictive in assessing the performance, as the worst cases characterized in [11] and [13] are of this form. The value of the resources is randomly selected according to a uniform distribution with minimum and maximum value of 1 and 100, respectively. For every instance generated, we randomly selected 50 different
initial allocations used as starting points for the algorithms.

The following table presents the results.

| Algorithm          | Average Poa | Average min Poa |
|--------------------|-------------|-----------------|
| BR with $f_{k_u}$  | 0.9701      | 0.9517          |
| BR with $f_{k}$    | 0.9723      | 0.9533          |
| Algorithm 1        | 0.9734      | 0.9538          |

In the first column we show the mean price of anarchy, i.e. the quantity $\frac{W(a_e)}{W(a_o)}$ averaged over all the instances and all the initial conditions. The second column presents the average-minimum price of anarchy, that is for every covering problem we compute the minimum of $\frac{W(a_e)}{W(a_o)}$ over all the initial conditions, and we then take the average over the instances.

We note that Algorithm 1 as well as the distribution rules $f_{k_u}$, $f_{k}$ perform similarly, when looking at a randomly generated instance. The efficiency values $\frac{W(a_e)}{W(a_o)}$ obtained are much higher compared to the analytical worst case, hinting at the fact that such instances are very few. Given that the average performance is similar, but the distribution $f_{k_u}$ is proven to have inferior worst case performance (Theorem 1), one might want to use either the optimal distribution $f_{k}$ or Algorithm 1. Recall indeed that the worst case performance of Algorithm 1 is on par or better to $f_{k}$ (Theorem 2). Nevertheless, the use of $f_{k}$ require knowledge of the cardinality $k$, while the algorithm proposed does not.

To conclude: Algorithm 1 achieves similar average performances compared to the other distributions tested, but has a better worst case performance than $f_{k_u}$ and a better-equal worst case performance than $f_{k}$ even if it does not require the knowledge of $k$.

VI. CONCLUSION

In this work we studied how additional information impacts the optimal design of local utility functions, when the goal is to improve the overall efficiency of a class of multiagent systems. Relative to covering problems, in the first part of the manuscript we highlighted an inherent tradeoff between potential risks and rewards when such additional information is uncertain. In the second part, we showed how it is possible to fully eliminate the risks by using a distributed algorithm that dynamically updated the local utility functions. The methodology used suggests that similar results could be obtained for a broader class of resource allocation problems than the one studied here.
APPENDIX A

PROOF OF PROPOSITION 2

Before proving the proposition, we report the expression for $\chi(f'_{k_d}, k_u)$, appearing in Equation (9)

$$
\chi(f'_{k_d}, k_u) = \frac{(k_u - 1)(k_u - 1)!}{k_u + (k_u - 1)\sum_{h=1}^{k_u-1-k_d} \frac{(k_u-1)!}{(k_u-h-1)!}(k_d-1)!}. 
$$

Thanks to result i) in Proposition 1 maximising $\text{PoA}(f, k_u)$ is equivalent to minimising $\chi(f, k_u)$ and $f'_{k_d}$ can be computed by the following linear program (LP) in the unknowns $x$, $\{f(j)\}_{j=1}^{k_u}$

$$
\min_{x \geq 0, f \in \mathcal{F}} x \\
\text{s.t.} \quad jf(j) - f(j + 1) \leq x \quad j \in [k_u - 1], \\
(k_u - 1)f(k_u) \leq x, \\
f(j) = f'_{k_d}(j) \quad j \in [k_d].
$$

We remove the constraints $x \geq 0$, $f \in \mathcal{F}$ as well as $jf(j) - f(j + 1) \leq x$ for $j \in [k_d - 1]$ and introduce the following relaxed linear program

$$
\min_{x, f} x \\
\text{s.t.} \quad jf(j) - f(j + 1) \leq x \quad j \in [k_d, k_u - 1], \\
(k_u - 1)f(k_u) \leq x, \\
f(j) = f'_{k_d}(j) \quad j \in [k_d].
$$

The proof is divided in two subproofs:

i) We show that a solution to the relaxed program (15) is given by (9) and (13).

ii) We show that the solution to the relaxed program obtained in i) is feasible for the original problem too.

Proof. i) The proof proceeds by showing that a solution of (15) can be obtained transforming all the inequality constraint into equalities. This will produce the expressions (9) and (13).

Let us define $v_j = f(j)$ for $j \in [p+1, k_u]$ and introduce the cost function $J(x, v_{k_d+1}, \ldots, v_{k_u}) = x$. We further introduce the constraint functions $g_1(x, v_{k_d+1}) = -x - v_{k_d+1}$ and $g_i(x, v_{p+i-1}, v_{p+i}) = -x + j v_{p+i-1} - v_{p+i}$ for $i \in [2, k_u - p]$ and $g_{k_u-k_d+1}(x, v_{k_u}) = -x + (k_u - 1)v_{k_u}$. With these
definitions the LP (15) is equivalent to the following where we have removed the decision
variables that are already determined

\[
\begin{align*}
\min_{x, v_{k_d+1}, \ldots, v_{k_u}} & \ J(x, v_{k_d+1}, \ldots, v_{k_u}), \\
\text{s.t.} & \ g_1(x, v_{k_d+1}) \leq -k_d f_{k_d}^*(k_d), \\
& \ g_i(x, v_{p+i-1}, v_{p+i}) \leq 0 \quad i \in [2, k_u - p], \\
& \ g_{k_u-k_d+1}(x, v_{k_u}) \leq 0.
\end{align*}
\]

Thanks to the convexity of the cost function and to the polytopic constraints, the Karush-Kuhn-Tucker conditions are necessary and sufficient for optimality [27]. Consequently, a feasible point \((x^*, v_{k_d+1}^*, \ldots, v_{k_u}^*)\) is an optimiser iff there exists \(\mu_i \in \mathbb{R}\) so that

\[
\nabla J^* + \sum_{i=1}^{k_u-k_d+1} \mu_i \nabla g_i^* = 0
\]

\[
g_i^* \leq 0, \quad \mu_i \geq 0, \quad \mu_i g_i^* = 0 \quad i \in [k_u - k_d + 1]
\]

where we used \(\nabla J^*\) to indicate \(\nabla J(x^*, v_{k_d+1}^*, \ldots, v_{k_u}^*)\), and similarly for \(g_i^*, \nabla g_i^*\). Observe that the distribution rule in (9) and the corresponding \(\chi(f_{k_d}', k_u)\) in (13) are the unique solution of the linear system \(g_i^* = 0\) for all \(i \in [k_u - k_d + 1]\), that is

\[
\begin{align*}
\begin{cases}
\begin{align*}
 j f_{k_d}'(j) - f_{k_d}'(j+1) - \chi(f_{k_d}', k_u) &= 0 \quad j \in [k_d, k_u - 1], \\
 (k_u - 1) f_{k_d}'(k_u) - \chi(f_{k_d}', k_u) &= 0, \\
 f_{k_d}'(j) &= f_{k_d}'(j) \quad j \in [k_d].
\end{align*}
\end{cases}
\end{align*}
\]

Primal feasibility and complementarity slackness are hence naturally satisfied. We are only left to prove that there exists \(\mu_i \geq 0\) such that \(\nabla J^* + \sum_{i=1}^{k_u-k_d+1} \mu_i \nabla g_i^* = 0\). We proceed by writing the stationarity conditions explicitly and show that this is indeed the case. Note that both the cost function and the constraints are linear so that their derivatives are constant functions

\[
\nabla J^* = (1, 0, \ldots, 0)
\]

\[
\nabla g_1^* = (-1, -1, 0, \ldots, 0)
\]

\[
\nabla g_2^* = (-1, k_d + 1, -1, 0, \ldots, 0)
\]

\[
\vdots
\]

\[
\nabla g_{k_u-k_d-1}^* = (-1, 0, \ldots, 0, k_u - 2, -1, 0)
\]

\[
\nabla g_{k_u-k_d}^* = (-1, 0, \ldots, 0, k_u - 1, -1)
\]

\[
\nabla g_{k_u-k_d+1}^* = (-1, 0, \ldots, 0, k_u - 1)
\]
Solving the stationarity condition in a recursive fashion starting from last component gives
\[
\begin{cases}
\mu_i = \mu_{k_u-k_d+1} \frac{(k_u-1)(k_u-1)!}{(k_u+i-1)!} & i \in [k_u - k_d] \\
\sum_{i=1}^{k_u-k_d+1} \mu_i = 1.
\end{cases}
\]
Substituting the first equation into the second one and solving yields
\[
\begin{cases}
\mu_i = \left( \sum_{i=1}^{k_u-k_d-1} \frac{(k_u-1)(k_u-1)!}{(k_u+i-1)!} \right)^{-1} \frac{(k_u-1)(k_u-1)!}{(k_u+i-1)!} & i \in [k_u - k_d], \\
\mu_{k_u-k_d+1} = \left( \sum_{i=1}^{k_u-k_d-1} \frac{(k_u-1)(k_u-1)!}{(k_u+i-1)!} \right)^{-1}.
\end{cases}
\]
Since \( \mu_i \geq 0 \) for all \( i \in [k_u - k_d + 1] \), we conclude that (9) and (13) solve the relaxed program (15).

Proof. ii) The proof proceeds by showing that (9) and (13) satisfy the constraints removed when transforming the original program (14) into (15).

Using (13) and (9), it is trivial to verify that \( \chi(f^*_{k_d}, k_u) \geq 0 \) and \( f^*_{k_d}(1) = 1 \), \( f^*_{k_d}(j) \geq 0 \). We proceed to prove that \( f_k^* \) is non increasing. Note that for \( j \leq k_d \), \( f^*_{k_d} \) coincides with \( f^*_{k_d} \), which was proven to be non increasing in [11]. Further, from Lemma 2 we know that \( j f'_{k_d}(j) - (j+1) f'_{k_d}(j+1) \geq 0 \) for \( j \in [k_d, k_u - 1] \). Thus
\[
j f'_{k_d}(j) - j f'_{k_d}(j+1) \geq j f'_{k_d}(j) - (j+1) f'_{k_d}(j+1) \geq 0
\]
for \( j \in [k_d, k_u - 1] \), which guarantees that \( f^*_{k_d} \) is non increasing for \( j \in [k_d, k_u] \) too.

We are left to show that \( j f'_{k_d}(j) - f'_{k_d}(j+1) \leq \chi(f^*_{k_d}, k_u) \) for \( j \in [k_d - 1] \). Since \( j \in [k_d - 1] \), it holds that
\[
j f'_{k_d}(j) - f'_{k_d}(j+1) = j f'_{k_d}(j) - f^*_{k_d}(j+1).
\]
Note that \( j f^*_{k_d}(j) - f^*_{k_d}(j+1) \leq \chi(f^*_{k_d}, k_d) \) by definition of \( \chi(f^*_{k_d}, k_d) \) in (4). Further, \( \chi(f^*_{k_d}, k_d) \leq \chi(f^*_{k_d}, k_u) \) for any \( k_d \leq k_u \) since the price of anarchy is a monotonically decreasing function (Proposition 1). Finally, Lemma 3 shows that for any \( k_d \leq k_u \), it holds \( \chi(f^*_{k_d}, k_u) \leq \chi(f^*_{k_d}, k_u) \). Hence \( j f'_{k_d}(j) - f'_{k_d}(j+1) \leq \chi(f^*_{k_d}, k_u) \) for \( j \in [k_d - 1] \). It follows that \( f^*_{k_d} \) is feasible for the original problem (14).

Thanks to this, and to the fact that \( f^*_{k_d} \) is optimal for (15), we conclude that \( f^*_{k_d} \) is a solution of the original problem. 

PROOF OF LEMMA 1

Proof. The result of Lemma 2 implies that for all \( k_d \leq k_u \)
\[
k_d f'_{k_d}(k_d) - j f'_{k_d}(j) \geq 0 \quad \forall j \in [k_d, k_u - 1].
\]

October 23, 2017 DRAFT
Further we know from [11] that \( k_d f_{k_d}^*(k_d) \leq 1 \). Since \( f_{k_d}^*(k_d) = f_{k_d}'(k_d) \), we conclude that for \( j \in [k_d, k_u - 1] \)

\[
1 \geq k_d f_{k_d}^*(k_d) = k_d f_{k_d}'(k_d) \geq j f_{k_d}'(j).
\]

For \( j \in [k_d] \), it holds \( f_{k_d}'(j) = f_{k_d}^*(j) \) and we already know that the optimal distribution \( f_{k_d}' \) does not overpay the players [11]. This concludes the proof.

\[
\Box
\]

**Proof of Theorem 1**

*Proof.* i) Thanks to Proposition 1, the performance of \( f_{k_u}^* \) on the class of games with cardinality \( k \) can be computed as \( \text{PoA}(f_{k_u}^*, k) = \frac{1}{1 + \chi(f_{k_u}^*, k)} \). Since \( k \leq k_u \), we can apply part i) of Lemma 4 to \( \chi(f_{k_u}^*, k) \) and conclude that

\[
\text{PoA}(f_{k_u}^*, k) = \frac{1}{1 + \chi(f_{k_u}^*, k_u)}.
\]

This means that the performance of \( f_{k_u}^* \) on the set of games with cardinality \( k \) is the same performance of the distribution \( f_{k_u}' \) on the set of games with cardinality \( k_u \geq k \), and

\[
\text{PoA}(f_{k_u}^*, k) = \text{PoA}(f_{k_u}^*, k_u) \leq \text{PoA}(f_k^*, k),
\]

where the last inequality holds since \( \text{PoA}(f_k^*, k) \) is a decreasing function of \( k \) as seen in part iii) of Proposition 1. The inequality is tight if and only if \( k = k_u \).

ii) Consider \( k_d \in [k_u - 1] \). The performance of \( f_{k_d}' \) on the class of games with cardinality \( k \) can be computed as \( \text{PoA}(f_{k_d}', k) = \frac{1}{1 + \chi(f_{k_d}', k)} \). Since \( k_d \in [k - 1] \), we apply part ii) of Lemma 4 to conclude that

\[
\text{PoA}(f_{k_d}', k) = \frac{1}{1 + \chi(f_{k_d}', k_u)}.
\]

Hence, for \( k_d \in [k_u - 1] \), the performance of \( f_{k_d}' \) in the class of games with cardinality \( k \) is the same of the performance in the class of games with cardinality \( k_u \) i.e. \( \text{PoA}(f_{k_d}', k) = \text{PoA}(f_{k_d}', k_u) \). Finally, by Lemma 3 we conclude that such performance is worse than what \( f_{k_u}^* \) can offer

\[
\text{PoA}(f_{k_d}', k) < \frac{1}{1 + \chi(f_{k_d}', k_u)} = \text{PoA}(f_{k_d}', k_u).
\]

Consider \( k_d \in [k, k_u] \). Since \( k \leq k_d \), only the first \( k \) entries of \( f_{k_d}' \) will determine the performance and these are identical to \( f_{k_d}' \) by definition of \( f_{k_d}' \). Hence \( \text{PoA}(f_{k_d}', k) = \frac{1}{1 + \chi(f_{k_d}', k)} \).

Further \( k_d \in [k, k_u] \) and part i) of Lemma 4 applies

\[
\text{PoA}(f_{k_d}', k) = \frac{1}{1 + \chi(f_{k_d}', k_d)} = \text{PoA}(f_{k_d}', k_d),
\]

so that \( f_{k_d}' \) has the same performance of \( f_{k_d}^* \). Using the fact that the optimal price of anarchy is a decreasing function, for any \( p \in [k, k_u] \) we get

\[
\text{PoA}(f_{k_d}', k) = \text{PoA}(f_{k_d}', k_d) \geq \text{PoA}(f_{k_u}^*, k_u) = \text{PoA}(f_{k_u}^*, k).
\]

The inequality is tight if and only if \( p = k_u \). 

\[
\Box
\]
PROOF OF THEOREM 2

Proof. i) Consider $x_r(t)$ for fixed $r \in \mathcal{R}$. The integer sequence $\{x_r(t)\}_{t=1}^{\infty}$ is upper bounded by the true cardinality $k$ (by definition of cardinality) and is non decreasing in $t$ thanks to its update rule (line 6 in Algorithm 1). Hence, after a finite number of steps, $x_r(t)$ has converged to $x_r^\infty$. Repeating the same reasoning for all the resources $r \in \mathcal{R}$ shows that the map $x_r(t)$ converges in a finite number of steps to a Nash equilibrium of the game with resource specific distribution rules fixed to their strategy. Since for $t \geq \hat{t}$ agents are playing round-robin best response on a potential game, their strategy will converge in a finite number of steps to a Nash equilibrium of the game with resource specific distribution rules fixed to $f^\text{alg}_{x_r}$ for $r \in \mathcal{R}$.

ii) Let us define $k_e = \max_{r \in \mathcal{R}} |a_e|_r$ (note that in general $k_e \neq k_M$). To ease the notation, in the following we will simply use $f(x_r,|a_r|)$ to indicate $f^\text{alg}_{x_r}(|a_r|)$, and similarly $u^i(a^i, a^{-i})$ to refer to $u^i(a^i, a^{-i}; \{x^\infty_r\}_{r \in \mathcal{R}})$. Further, we define $A_e = \bigcup_i a^i_e$ and $A_o = \bigcup_i a^i_o$.

By definition of equilibrium we have for all $i \in [n]$, $u^i(a^i_e, a^{-i}_e) \geq u^i(a^i_o, a^{-i}_e)$ and hence

$$0 \leq \sum_{i \in [n]} u^i(a^i_e, a^{-i}_e) - \sum_{i \in [n]} u^i(a^i_o, a^{-i}_e).$$

Using the definition of payoff, the first term can be rewritten as

$$\sum_{i \in [n]} u^i(a^i_e, a^{-i}_e) = \sum_{i \in [n]} \sum_{r \in a^i_e} f(x_r, |a_r|) v_r$$

$$= \sum_{r \in A_e} |a_e|_r f(x_r, |a_r|) v_r = \sum_{j=1}^{k_e} \sum_{r \in A_e |a_e|_r = j} j f(x_r, j) v_r. \tag{18}$$

With a similar manipulation the second term becomes

$$\sum_{i \in [n]} u^i(a^i_o, a^{-i}_e) = \sum_{i \in [n]} \sum_{r \in a^i_o} f(x_r, |(a^i_o, a^{-i}_e)|_r) v_r$$

$$\geq \sum_{i \in [n]} \sum_{r \in a^i_o} f(x_r, \min\{k, |a_r| + 1\}) v_r,$$

this holds because for all resources $|(a^i_o, a^{-i}_e)|_r \leq |a_e|_r + 1$, $|(a^i_o, a^{-i}_e)|_r \leq k$ and $f$ is non increasing in its second argument. For resources $r \in a_o$ it holds $|a_o|_r \geq 1$, and so

$$\sum_{i \in [n]} \sum_{r \in a^i_o} f(x_r, \min\{k, |a_r| + 1\}) v_r$$

$$= \sum_{r \in A_o} |a_o|_r f(x_r, \min\{k, |a_r| + 1\}) v_r$$

$$\geq \sum_{r \in A_o} f(x_r, \min\{k, |a_r| + 1\}) v_r.$$
The second term in (17) can thus be lower bounded by
\[
\sum_{i \in [n]} u^i(a^i_o, a^i_e) \geq \sum_{r \in A_o} f(x_r, \min\{k, |a_e|_r + 1\}) v_r
\]
\[
= \sum_{j=0}^{k_e} \sum_{r \in A_o, |a_e|_r = j} f(x_r, \min\{k, j + 1\}) v_r.
\] (19)

Substituting (18) and (19) in (17) gives
\[
0 \leq \sum_{i \in [n]} u^i(a^i_o, a^i_e) - \sum_{i \in [n]} u^i(a^i_o, a^i_e)
\]
\[
\leq \sum_{j=1}^{k_e} \sum_{r \in A_o, |a_e|_r = j} j f(x_r, j) v_r - \sum_{j=0}^{k_e} \sum_{r \in A_o, |a_e|_r = j} f(x_r, \min\{k, j + 1\}) v_r
\]
\[
= \sum_{j=1}^{k_e} \sum_{r \in A_o, |a_e|_r = j} j f(x_r, j) v_r - \sum_{j=1}^{k_e} \sum_{r \in A_o, |a_e|_r = j} f(x_r, \min\{k, j + 1\}) v_r
\]
\[
- \sum_{r \in A_o \setminus A_e} v_r f(x_r, 1)
\]
\[
= \sum_{j=1}^{k_e} \sum_{r \in A_o \setminus A_o} j f(x_r, j) v_r + \sum_{j=1}^{k_e} \sum_{r \in A_o \cap A_o, |a_e|_r = j} f(x_r, j) v_r
\]
\[
- \sum_{j=1}^{k_e} \sum_{r \in A_o, |a_e|_r = j} f(x_r, \min\{k, j + 1\}) v_r - \sum_{r \in A_o \setminus A_e} v_r f(r, 1)
\]
\[
= \sum_{j=1}^{k_e} \sum_{r \in A_o \setminus A_o} j f(x_r, j) v_r - \sum_{r \in A_o \setminus A_e} v_r
\]
\[
+ \sum_{j=1}^{k_e} \sum_{r \in A_o, |a_e|_r = j} \left( j f(x_r, j) - f(x_r, \min\{k, j + 1\}) \right) v_r,
\] (20)

where we have used the fact that \( f(x_r, 1) = 1 \) for all resources. We intend to bound the first and the third term in the last expression. In the summands of (20) \( j = |a_e|_r \leq x_r^\infty \) due to the update of \( x_r(t) \) in Algorithm 1 and recall that \( f(x_r, |a|_r) = f^{ALG}_{x_r^\infty}(|a|_r). \) Hence we can apply Lemma 5 to the first term in (20)
\[
\sum_{j=1}^{k_e} \sum_{r \in A_o \setminus A_o, |a_e|_r = j} j f(x_r, j) v_r \leq \sum_{j=1}^{k_e} \sum_{r \in A_o \setminus A_o, |a_e|_r = j} (\chi(f^*_r M, k_M) + 1) v_r.
\] (21)
Similarly for the third term in (20)
\[
\sum_{j=1}^{k_e} \sum_{r \in A_e \setminus A_o \atop |a_e)_r = j} \left( jf(x_r, j) - f(x_r, \min\{k, j + 1\}) \right) v_r
\]
\[
\leq \sum_{j=1}^{k_e} \sum_{r \in A_o \setminus A_e \atop |a_e)_r = j} \chi(f_{k_M}^*, k_M) v_r = \chi(f_{k_M}^*, k_M) \sum_{r \in A_e \cap A_o} v_r.
\]

Hence combining (20) with the bounds from (21) and (22)
\[
0 \leq (\chi(f_{k_M}^*, k_M) + 1) \sum_{j=1}^{k_e} \sum_{r \in A_e \setminus A_o \atop |a_e)_r = j} v_r - \sum_{r \in A_o \setminus A_e} v_r
\]
\[
+ (\chi(f_{k_M}^*, k_M) + 1) \sum_{r \in A_e \setminus A_o} v_r - \sum_{r \in A_o \setminus A_e} v_r
\]
\[
= (\chi(f_{k_M}^*, k_M) + 1) \sum_{r \in A_e \setminus A_o} v_r - \sum_{r \in A_o \setminus A_e} v_r
\]
\[
+ (\chi(f_{k_M}^*, k_M) + 1) \sum_{r \in A_e \setminus A_o} v_r - \sum_{r \in A_o \setminus A_e} v_r
\]
\[
= (\chi(f_{k_M}^*, k_M) + 1) \sum_{r \in A_e \setminus A_o} v_r - \sum_{r \in A_o \setminus A_e} v_r
\]
\[
= (\chi(f_{k_M}^*, k_M) + 1) W(a_e) - W(a_o).
\]
Hence \((\chi(f_{k_M}^*, k_M) + 1) W(a_e) - W(a_o) \geq 0\) and rearranging
\[
\frac{W(a_e)}{W(a_o)} \geq \frac{1}{1 + \chi(f_{k_M}^*, k_M)} = \text{PoA}(f_{k_M}^*, k_M) \geq \text{PoA}(f_k^*, k),
\]
where the last inequality follows from the fact that the price of anarchy is a decreasing function, and \(k_M \leq k\) by definition of cardinality.

\[\square\]

**Proof of Theorem 3**

**Proof.** i) Consider the covering problem depicted in the following figure (a), composed of players \(p_1, p_2, p_3\) represented by a solid dot; resources \(r_1, r_2, r_3, r_4\) represented by a circle with values \(v_1, v_2, v_3, v_4\) such that
\[
v_1 > v_3 > v_4 > v_2 \quad \text{and} \quad v_1 f_3^*(2) < v_2 < v_1 f_2^*(2) < v_4
\]
As an example take \( v = (11, 5, 7, 6) \). Each player \( p_1, p_2, p_3 \) can choose only one resource from \( \{r_1, r_2, r_3\}, \{r_2, r_3, r_4\}, \{r_1, r_2, r_3, r_4\} \), respectively i.e. each player can only choose one arrow pointing outwards from himself.

The cardinality is \( k = 3 \) since all players could choose simultaneously \( r_2 \) or \( r_3 \), hence the optimal distribution rule is \( f_3^* \). Amongst the equilibria obtained with \( f_3^* \) there is \( a_e = (r_2, r_3, r_1) \), depicted in the previous figure (b). This configuration is an equilibrium since \( v_2 > v_1 f_3^*(2) > v_3 f_3^*(2) \) and \( v_3 > v_4, v_3 > v_2 f_3^*(2) \) and \( v_1 > v_2 f_3^*(2), v_1 > v_3 f_3^*(2), v_1 > v_4 \). Such equilibrium gives a welfare of \( v_1 + v_2 + v_3 \) that is less than the optimal \( v_1 + v_3 + v_4 \), since \( v_2 < v_4 \). We intend to show that for any initial condition and for any execution, Algorithm 1 will converge to an optimal allocation. This suffices to prove that the worst equilibrium obtained with Algorithm 1 performs better than the worst equilibrium obtained with \( f_3^* \), which is not optimal as shown above. Observe that the conditions \( v_4 > v_3 > v_2 > v_1 \) and \( v_4 > v_1 f_2^*(2) \) ensure that an allocation with two or more agents covering the same resource is never an equilibrium. This holds regardless of the distribution used. Hence, the welfare can potentially take \( \binom{4}{3} = 4 \) different values, since the binomial represents the number of subsets with 3 elements (agents allocations) that can be extracted from a set of 4 elements (set of resources). These different welfare values are obtained for \( (r_1, r_2, r_4), (r_2, r_3, r_4), (r_1, r_3, r_4), (r_1, r_2, r_3) \), or feasible permutations of each. The allocation \( (r_1, r_2, r_4) \) is never an equilibrium since player \( p_3 \) can improve moving to \( r_3 \) because \( v_3 > v_4 \). Similarly for any feasible permutation of \( (r_1, r_2, r_4) \), the player selecting resource \( r_4 \) can always improve moving to \( r_3 \). The allocation \( (r_2, r_3, r_4) \) is never an equilibrium since player \( p_3 \) can improve moving to \( r_1 \) since \( v_1 \) is the highest. Similarly for any feasible permutation of \( (r_2, r_3, r_4) \), there exists a player that can improve moving to \( r_1 \). This holds regardless of what distribution rule is used. The allocation \( (r_1, r_3, r_4) \) (or any feasible permutation) is optimal. We are thus left to show that Algorithm 1 never converges to \( (r_1, r_2, r_3) \), or any other feasible permutation. We show this by enumeration.
The allocation \((r_1, r_2, r_3)\) can not be an equilibrium since player \(p_2\) can improve moving to \(r_4\) because \(v_4 > v_2\). The allocation \((r_1, r_3, r_2)\) can not be an equilibrium since player \(p_3\) can improve moving to \(r_4\). The allocation \((r_3, r_2, r_1)\) can not be an equilibrium since player \(p_2\) can improve moving to \(r_4\). We are left to check \(a_e = (r_2, r_3, r_1)\), depicted in the previous figure (b). This can not be an equilibrium of Algorithm 1, because \(v_2 < v_1^a_l(2)\) for \(l = 1, 2\) and so player \(p_2\) could improve moving to \(r_1\). The fact that the algorithm uses \(l \leq 2\) on resource \(r_1\) holds because the maximum number of players on \(r_1\) is two, and so \(k_i(1) \leq 2\) at any time step \(t \in [n]\). We conclude that all the equilibria towards which the algorithm converges give optimal welfare, while \(f_3^*\) also produces the suboptimal equilibrium \(a_e\); the claim follows. Observe that this is not a worst case instance because the price of anarchy with the example values \(v = (11, 5, 7, 6)\) is

\[
\frac{W(a_e)}{W(a_o)} = \frac{11 + 5 + 7}{11 + 6 + 7} = \frac{23}{24} > \text{PoA}(f_3^*, 3) = \frac{7}{11}.
\]

i) Consider the covering problem depicted in the following figure (a), composed of players \(p_1, p_2, p_3\) represented by a solid dot; resources \(r_1, r_2, r_3\) represented by an empty circle with values \(v_1, v_2, v_3\) such that

\[
v_3 f_3^*(2) < v_1 < v_2 < v_3/2 < v_3.
\]

As an example take \(v = [9, 9.5, 20]\). Each player \(p_1, p_2, p_3\) can choose only one resource from \(\{r_1, r_2\}, \{r_2, r_3\}, \{r_1, r_2, r_3\}\), respectively i.e. each player can only choose one arrow pointing outwards from himself.

The cardinality is \(k = 3\) since all players could choose simultaneously \(r_1\), hence the optimal distribution rule is \(f_3^*\). All the equilibria obtained with \(f_3^*\) are completely spread i.e. they feature one and only one player on each resource. Any allocation where there are two or more players in one resource is not an equilibrium for \(f_3^*\), as detailed in the following. If all three players selected resource \(r_2\), \(p_2\) could improve moving to \(r_3\) since \(v_3 > v_2\). If \(p_1\) and \(p_3\) selected \(r_1\), depending on the choice of \(p_2\), either \(p_1\) or \(p_3\) could improve moving respectively to \(r_2\) or \(r_3\) since \(v_2 > v_1\) and \(v_3 > v_1\). If \(p_2\) and \(p_3\) selected \(r_3\), depending on the choice...
of \( p_1 \), either \( p_2 \) or \( p_3 \) could improve moving respectively to \( r_2 \) or \( r_1 \) since \( v_3 f_3^*(2) < v_2 \) and \( v_3 f_3^*(2) < v_1 \). If \( p_1, p_3 \) selected both \( r_2 \), regardless of the choice of \( p_2, p_3 \) could improve moving to \( r_3 \) since \( v_3 > v_2 \). If \( p_2, p_3 \) selected both \( r_2 \), regardless of the choice of \( p_1, p_3 \) could improve moving to \( r_3 \) since \( v_3 > v_2 \). Finally, if \( p_1, p_2 \) selected \( r_2 \), regardless of the choice of \( p_3, p_2 \) could always improve moving to \( r_3 \) since \( v_3 > v_2 \). Thus all equilibria obtained with \( f_3^* \) (including the worst) give a welfare of \( v_1 + v_2 + v_3 \).

Let us consider Algorithm 1 and initialise it at \( a_1 = [r_2, r_3, r_1] \), giving \( k_1(r) = 1 \) for all \( r \). Player \( p_3 \) updates and since \( v_3 \cdot 1 > v_1 \cdot 1 \), he selects \( r_3 \), giving \( a_2 = [r_2, r_3, r_3] \) and \( k_2(r) = 1 \) for \( r_1, r_2 \) and \( k_2(r_3) = 2 \). This allocation is depicted in the previous figure (b) and is an equilibrium configuration. Indeed \( p_1 \) can not improve since \( v_2 > v_1 \); \( p_2 \) can not improve since \( v_3 f_3^*(2) + \frac{v_3}{2} > v_2 \cdot 1 \); \( p_3 \) can not improve since \( v_3 f_2^*(2) + \frac{v_3}{2} > v_2 \cdot 1 \) and \( v_3 f_2^*(2) + \frac{v_3}{2} > v_1 \cdot 1 \). Such equilibrium has a welfare of \( v_2 + v_3 \).

In conclusion, all equilibria obtained with \( f_3^* \) give a better welfare than \( a_2 \) and thus of the worst equilibrium obtained with Algorithm 1.

\[ \square \]

**LEMMA**

**Lemma 2.** Let \( k_d \in [k_u] \). The distribution \( f_{k_d} \) satisfies

\[
jj'_{k_d}(j) - (j + 1)f'_{k_d}(j + 1) \geq 0 \quad j \in [k_d, k_u - 1].
\]

**Proof.** Recall that \( f_{k_d}' \) is obtained from equation (16). Using \( \chi(f_{k_d}', k_u) \) from (13), one can reconstruct the tail entries of \( f_{k_d}'(j) \) with the following backward recursion

\[
jj'_{k_d}(j) - f'_{k_d}(j + 1) = \chi(f_{k_d}', k_u) \quad j \in [k_d, k_u - 1],
\]

\[
(k_u - 1)f_{k_d}'(k_u) = \chi(f_{k_d}', k_u).
\]

Starting from \( f_{k_d}'(k_u) = \frac{\chi(f_{k_d}', k_u)}{k_u - 1} \), the first equation gives for \( j \geq k_d \)

\[
jj'_{k_d}(j) = \chi(f_{k_d}', k_u) \left( 1 + \sum_{i=j+1}^{k_u-1} \frac{j!}{i!} + \frac{j!}{(k_u - 1)(k_u - 1)!} \right).
\]

Hence

\[
\frac{1}{\chi(f_{k_d}', k_u)} (jj'_{k_d}(j) - (j + 1)f'_{k_d}(j + 1)) =
\]

\[
= \sum_{i=j+1}^{k_u-1} \frac{j!}{i!} - \sum_{i=j+2}^{k_u-1} \frac{(j + 1)!}{i!} + \frac{j! - (j + 1)!}{(k_u - 1)(k_u - 1)!} =
\]

\[
= \sum_{i=j+1}^{k_u-2} \left( \frac{j!}{i!} - \frac{(j + 1)!}{(i + 1)!} \right) + \frac{j!}{(k_u - 1)!} \left( 1 + \frac{1}{k_u - 1} - \frac{j + 1}{k_u - 1} \right).
\]
Note that for $i > j$, one has
\[
\frac{j!}{i!} = \frac{1}{i(i - 1) \ldots (j + 1)}\quad \text{and so } \frac{j!}{i!} - \frac{(j + 1)!}{(i + 1)!} > 0.
\]
Further
\[
1 + \frac{1}{k_u - 1} + \frac{j + 1}{k_u - 1} = \frac{k_u - j - 1}{k_u - 1} \geq 0
\]
since $j \leq k_u - 1$ by assumption. Hence we conclude that
\[
 j f'_{k_d}(j) - (j + 1) f'_{k_d}(j + 1) \geq 0 \quad j \in [k_d, k_u - 1].
\]
\[\square\]

**Lemma 3.** For any $1 < k_d < k_u$ it holds $\chi(f^*_u, k_u) < \chi(f'_u, k_u)$.

**Proof.** The expression of $\chi(f^*_u, k_u)$ in (6) and of $\chi(f'_u, k_u)$ in equation (13) can be rewritten as
\[
\chi(f^*_u, k_u) = \frac{(k_u - 1)(k_u - 1)!}{1 + (k_u - 1)(k_u - 1)! \sum_{i=1}^{k_u-1} \frac{1}{i!}},
\]
\[
\chi(f'_u, k_u) = \frac{(k_u - 1)(k_u - 1)!}{k_u + \sum_{h=1}^{k_u-1-k_d} \frac{(k_u - 1)(k_u - 1)!}{(k_u - h - 1)!} \beta(k_d)},
\]
where
\[
\beta(k_d) := \frac{1}{1 + \sum_{h=1}^{k_u-1-k_d} \frac{1}{k_u-1)(k_u-1)!}}.
\]
Instead of showing $\chi(f^*_u, k_u) < \chi(f'_u, k_u)$, in the following we equivalently prove that $\frac{1}{\chi(f^*_u, k_u)} > \frac{1}{\chi(f'_u, k_u)}$ i.e., that
\[
1 + (k_u - 1)(k_u - 1)! \sum_{i=1}^{k_u-1} \frac{1}{i!} = k_u + (k_u - 1)(k_u - 1)! \sum_{i=1}^{k_u-2} \frac{1}{i!} >
\]
\[
\left(k_u + \sum_{h=1}^{k_u-1-k_d} \frac{(k_u - 1)(k_u - 1)!}{(k_u - h - 1)!}\right) \left(1 + \sum_{h=1}^{k_u-1} \frac{(k_u - 1)(k_u - 1)!}{h!}\right).
\]
The previous inequality can be rewritten as
\[
(k_u - 1)(k_u - 1)!\left(\sum_{i=1}^{k_u-2} \frac{1}{i!} - \sum_{h=1}^{k_u-1-k_d} \frac{1}{(k_u - h - 1)!}\right) >
\]
\[
\left(k_u + \sum_{h=1}^{k_u-1-k_d} \frac{(k_u - 1)(k_u - 1)!}{(k_u - h - 1)!}\right) \left(\sum_{h=1}^{k_u-1} \frac{(k_u - 1)(k_u - 1)!}{h!}\right).
\]
Since the left hand side is equal to $(k_u - 1)(k_u - 1)! \sum_{h=1}^{k_u-1} \frac{1}{h!}$, we can simplify the term $\sum_{h=1}^{k_u-1} \frac{1}{h!}$ to get
\[
\frac{(k_u - 1)(k_u - 1)!}{(k_d - 1)(k_d - 1)!} > \left(k_u + \sum_{h=1}^{k_u-1-k_d} \frac{(k_u - 1)(k_u - 1)!}{(k_u - h - 1)!}\right).
\]
which is finally equivalent to
\[
\frac{1}{(k_d - 1)(k_d - 1)!} > \frac{k_u}{(k_u - 1)(k_u - 1)} + \sum_{h=1}^{k_u - k_d} \frac{1}{(k_u - h - 1)!}.
\] (26)

We use induction to show that inequality (26) holds for \(1 < k_d < k_u\), as required. We start from \(k_d = k_u - 1\) and apply induction backwards until we reach \(k_d = 2\).

i) For \(k_d = k_u - 1\) and \(k_d > 1\) inequality (26) reads as
\[
\frac{1}{(k_d - 1)(k_d - 1)!} > \frac{k_d + 1}{k_d \cdot k_d!} \iff k_d^2 > k_d^2 - 1,
\]
which is always satisfied.

ii) Let us assume the inequality holds for a generic \(k_d \leq k_u - 1\), we show that it holds also for \(k_d - 1\) (with \(k_d > 1\)). That is, we assume
\[
\frac{1}{(k_d - 1)(k_d - 1)!} > \frac{k_u}{(k_u - 1)(k_u - 1)} + \sum_{h=1}^{k_u - k_d} \frac{1}{(k_u - h - 1)!},
\] (27)
and want to show
\[
\frac{1}{(k_d - 2)(k_d - 2)!} > \frac{k_u}{(k_u - 1)(k_u - 1)} + \sum_{h=1}^{k_u - k_d} \frac{1}{(k_u - h - 1)!},
\] (28)

We can rewrite the right hand side of (28) and use (27) to upper bound it
\[
\frac{k_u}{(k_u - 1)(k_u - 1)} + \sum_{h=1}^{k_u - k_d} \frac{1}{(k_u - h - 1)!} = \frac{k_u}{(k_u - 1)(k_u - 1)} + \sum_{h=1}^{k_u - k_d - 1} \frac{1}{(k_u - h - 1)!} + \frac{1}{(k_d - 1)!} < \frac{1}{(k_d - 1)(k_d - 1)!} + \frac{1}{(k_d - 1)!} = \frac{k_d}{(k_d - 1)(k_d - 1)!} < \frac{1}{(k_d - 2)(k_d - 2)!}.
\] (29)

The last inequality holds since it is equivalent to
\[
\frac{k_d}{(k_d - 1)^2} < \frac{1}{k_d - 2} \iff k_d^2 - 2k_d < (k_d - 1)^2,
\]
which is always satisfied. Comparing the first and last term in (29) gives (28).

This completes the induction and thus the proof.

\[ \square \]

**Lemma 4.**  i) For any \(1 \leq l \leq m, m \in \mathbb{N}\) it holds
\[
\chi(f_m^*, l) = \chi(f_m^*, m).
\]
ii) For any \( k \in [k_u] \) and \( 1 < k_d < k \) it holds
\[
\chi(f'_{k_d}, k) = \chi(f'_{k_d}, k_u) .
\]

Proof. i) If \( l = m \), the result holds trivially. Hence in the following we consider \( l \in [m - 1] \). By definition of \( \chi(f_m^*, l) \) in (4), one has
\[
\chi(f_m^*, l) = \min_{x \geq 0} x \\
\text{s.t. } jf_m^*(j) - f_m^*(j + 1) \leq x \quad j \in [l - 1] \\
(l - 1)f_m^*(l) \leq x .
\]
Note that \( f_m^* \) is derived in [11, Theorem 2] solving the following recursion
\[
lf_m^*(l) - f_m^*(l + 1) = \chi(f_m^*, m) \quad j \in [m - 1] \\
(l - 1)f_m^*(l) = \chi(f_m^*, m) .
\]
Since \( m > l \), it follows that any feasible \( x \) from the LP above has to satisfy \( x \geq \chi(f_m^*, m) \). In the following we show that setting \( x = \chi(f_m^*, m) \), the constraint \( (l - 1)f_m^*(l) \leq x \) is also satisfied. This will be enough to conclude that \( \chi(f_m^*, l) = \chi(f_m^*, m) \).
Since \( f_m^* \) is non increasing, one has
\[
(l - 1)f_m^*(l) - \chi(f_m^*, m) = lf_m^*(l) - f_m^*(l) - \chi(f_m^*, m) \leq \\
\leq lf_m^*(l) - f_m^*(l + 1) - \chi(f_m^*, m) = 0 ,
\]
where the equality holds applying (30) for \( j = l \in [m - 1] \).
ii) We intend to compute
\[
\chi(f'_{k_d}, k) = \min_{x \geq 0} x \\
\text{s.t. } jf'_{k_d}(j) - f'_{k_d}(j + 1) \leq x \quad j \in [k - 1] \\
(k - 1)f'_{k_d}(k) \leq x .
\]
For any feasible \( x \), it must be \( x \geq \chi(f'_{k_d}, k_u) \) due to how \( f'_{k_d}(j) \) is recursively defined for \( j > k_d \) in Equation (16). Similarly to what shown before, one can prove that \( x = \chi(f'_{k_d}, k_u) \) will also satisfy the constraint \( (k - 1)f'_{k_d}(k) \leq x \). Hence \( \chi(f'_{k_d}, k) = \chi(f'_{k_d}, k_u) \) and the proof is concluded.

Lemma 5. For all resources \( r \in R \), the distribution rules \( f_{x_r}^{\text{alg}} \) are such that
\[
j f_{x_r}^{\text{alg}}(j) \leq \chi(f_{k_M}^*, k_M) + 1 \quad j \in [x_r^\infty] \tag{31}
\]
\[
j f_{x_r}^{\text{alg}}(j) - f_{x_r}^{\text{alg}}(\min\{k, j + 1\}) \leq \chi(f_{k_M}^*, k_M) \quad j \in [x_r^\infty] \tag{32}
\]
where \( k_M = \max_{r \in \mathcal{R}} x_r^\infty \).

**Proof.** We start from (31) and examine \( f_{x_r^\infty}^{\text{alg}} \) for a fixed \( r \in \mathcal{R} \). Consider \( j \in [x_r^\infty - 1] \), by definition of \( f_{x_r^\infty}^{\text{alg}} \) and the fact that \( f_{x_r^\infty}^* \) is non increasing

\[
j f_{x_r^\infty}^{\text{alg}}(j) - 1 = j f_{x_r^\infty}^*(j) - f_{x_r^\infty}^*(1)
\leq j f_{x_r^\infty}^*(j) - f_{x_r^\infty}^*(j + 1) \leq \chi(f_{x_r^\infty}^*, x_r^\infty)
\implies j f_{x_r^\infty}^{\text{alg}}(j) \leq \chi(f_{x_r^\infty}^*, x_r^\infty) + 1,
\]

where the last inequality holds thanks to the definition (4). Since \( x_r^\infty \leq k_M \) and the price of anarchy is a decreasing function, one has \( \chi(f_{x_r^\infty}^*, x_r^\infty) \leq \chi(f_{k_M}^*, k_M) \) and so \( j f_{x_r^\infty}^{\text{alg}}(j) \leq \chi(f_{k_M}^*, k_M) + 1 \) for \( j \in [x_r^\infty - 1] \).

In a similar fashion when \( j = x_r^\infty \)

\[
x_r^\infty f_{x_r^\infty}^{\text{alg}}(x_r^\infty) - 1 = x_r^\infty f_{x_r^\infty}^*(x_r^\infty) - f_{x_r^\infty}^*(1)
\leq x_r^\infty f_{x_r^\infty}^*(x_r^\infty) - f_{x_r^\infty}^*(x_r^\infty)
= (x_r^\infty - 1) f_{x_r^\infty}^*(x_r^\infty)
\leq \chi(f_{k_M}^*, x_r^\infty) \leq \chi(f_{k_M}^*, k_M),
\]

where the only difference is in the last equality that comes from equation (6). Repeating the same reasoning for all \( r \in \mathcal{R} \), one has proven (31).

In the remaining, we show that (32) holds. Consider \( f_{x_r^\infty}^{\text{alg}} \) for a fixed resource \( r \in \mathcal{R} \) and recall that \( x_r^\infty \leq k \). Thus for \( j \in [x_r^\infty - 1] \) one has \( \min\{k, j + 1\} = j + 1 \) and the claim reads as

\[
j f_{x_r^\infty}^{\text{alg}}(j) - f_{x_r^\infty}^{\text{alg}}(j + 1) = j f_{x_r^\infty}^*(j) - f_{x_r^\infty}^*(j + 1)
\leq \chi(f_{x_r^\infty}^*, x_r^\infty) \leq \chi(f_{k_M}^*, k_M)
\]

where the first inequality holds thanks to definition (4) and the last since the price of anarchy is non increasing (\( x_r^\infty \leq k_M \)).

In the remaining we focus on \( j = x_r^\infty \) and divide the proof in two subparts. When \( k = x_r^\infty \), \( \min\{k, j + 1\} = x_r^\infty \) and the claim follows from

\[
x_r^\infty f_{x_r^\infty}^{\text{alg}}(x_r^\infty) - f_{x_r^\infty}^{\text{alg}}(x_r^\infty) = x_r^\infty f_{x_r^\infty}^*(x_r^\infty) - f_{x_r^\infty}^*(x_r^\infty)
= (x_r^\infty - 1) f_{x_r^\infty}^*(x_r^\infty) \leq \chi(f_{k_M}^*, x_r^\infty) \leq \chi(f_{k_M}^*, k_M),
\]

similarly to (33). When \( k > x_r^\infty \), then \( \min\{k, j + 1\} = x_r^\infty + 1 \) and the claim holds if we show

\[
x_r^\infty f_{x_r^\infty}^{\text{alg}}(x_r^\infty) - f_{x_r^\infty}^{\text{alg}}(x_r^\infty + 1) \leq \chi(f_{x_r^\infty}^*, x_r^\infty),
\]
For this to hold, one has to require

\[
x^{\infty}_r f^{\text{alg}}(x^{\infty}_r + 1) \geq x^{\infty}_r f^{\text{alg}}(x^{\infty}_r) - \chi(f^{\star}, x^{\infty}_r)
\]

\[
= x^{\infty}_r f^{\star}(x^{\infty}_r) - \chi(f^{\star}, x^{\infty}_r)
\]

\[
= f^{\star}(x^{\infty}_r) = f^{\text{alg}}(x^{\infty}_r),
\]

where the second equality sign follows form \( \chi(f^{\star}, x^{\infty}_r) = (x^{\infty}_r - 1) f^{\star}(x^{\infty}_r) \), that is form equation (4). Hence we need to impose

\[
f^{\text{alg}}(x^{\infty}_r + 1) \geq f^{\text{alg}}(x^{\infty}_r),
\]

but at the same time we are limited to non increasing distribution rules. Hence we set \( f^{\text{alg}}(x^{\infty}_r + 1) = f^{\text{alg}}(x^{\infty}_r) \) as by definition of \( f^\ell \) from Equation (11). The proof is completed by observing that the same reasoning can be repeated for any resource \( r \in \mathcal{R} \).

\[ \square \]

REFERENCES

[1] D. Paccagnan and J. R. Marden, “The risks and rewards of conditioning noncooperative designs to additional information,” in Proceedings of the 55th Allerton Conference on Communication, Control, and Computing, 2017.

[2] D. D. Siljak, Decentralized control of complex systems. Courier Corporation, 2011.

[3] D. Paccagnan, M. Kamgarpour, and J. Lygeros, “On aggregative and mean field games with applications to electricity markets,” in 2016 European Control Conference (ECC), June 2016, pp. 196–201.

[4] P. N. Brown and J. R. Marden, “Studies on robust social influence mechanisms: Incentives for efficient network routing in uncertain settings,” IEEE Control Systems, vol. 37, no. 1, pp. 98–115, Feb 2017.

[5] J. R. Marden and A. Wierman, “Distributed welfare games,” Operations Research, vol. 61, no. 1, pp. 155–168, 2013.

[6] S. Martinez, J. Cortes, and F. Bullo, “Motion coordination with distributed information,” IEEE Control Systems, vol. 27, no. 4, pp. 75–88, Aug 2007.

[7] M. Pavone, A. Arsie, E. Frazzoli, and F. Bullo, “Distributed algorithms for environment partitioning in mobile robotic networks,” IEEE Transactions on Automatic Control, vol. 56, no. 8, pp. 1834–1848, Aug 2011.

[8] C. Langbort, R. S. Chandra, and R. D’Andrea, “Distributed control design for systems interconnected over an arbitrary graph,” IEEE Transactions on Automatic Control, vol. 49, no. 9, pp. 1502–1519, Sept 2004.

[9] B. Bamieh, F. Paganini, and M. A. Dahleh, “Distributed control of spatially invariant systems,” IEEE Transactions on Automatic Control, vol. 47, no. 7, pp. 1091–1107, Jul 2002.

[10] J. R. Marden, “The role of information in multiagent coordination,” in 53rd IEEE Conference on Decision and Control, Dec 2014, pp. 445–450.

[11] M. Gaizinger, Covering Games: Approximation through Non-cooperation. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 184–195.

[12] C. Langbort and V. Gupta, “Minimal interconnection topology in distributed control design,” SIAM Journal on Control and Optimization, vol. 48, no. 1, pp. 397–413, 2009.

[13] V. Ramaswamy, D. Paccagnan, and J. R. Marden, “The impact of local information on the performance of multiagent systems,” ArXiv:1710.01409, 2017.

[14] V. V. Vazirani, Approximation Algorithms. New York, NY, USA: Springer-Verlag New York, Inc., 2001.
[15] U. Feige, “A threshold of ln n for approximating set cover,” J. ACM, vol. 45, no. 4, pp. 634–652, July 1998.

[16] R. A. Murphey, Target-Based Weapon Target Assignment Problems. Boston, MA: Springer US, 2000, pp. 39–53.

[17] U. Feige and J. Vondrak, “Approximation algorithms for allocation problems: Improving the factor of 1 - 1/e,” in 2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS’06), Oct 2006, pp. 667–676.

[18] H.-L. Chen, T. Roughgarden, and G. Valiant, “Designing network protocols for good equilibria,” SIAM Journal on Computing, vol. 39, no. 5, pp. 1799–1832, 2010.

[19] B. Gentile, F. Parise, D. Paccagnan, M. Kamgarpour, and J. Lygeros, “Nash and wardrop equilibria in aggregative games with coupling constraints,” arXiv preprint arXiv:1702.08789, 2017.

[20] G. Nemhauser, L. Wolsey, and M. Fisher, “An analysis of approximations for maximizing submodular set functions – I,” Mathematical Programming, vol. 14, no. 1, pp. 265–294, 1978.

[21] A. Krause and C. Guestrin, “Near-optimal observation selection using submodular functions,” in Proceedings of the 22nd Conference on Artificial Intelligence, 2007.

[22] J. R. Marden, G. Arslan, and J. S. Shamma, “Cooperative control and potential games,” IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 39, no. 6, pp. 1393–1407, Dec 2009.

[23] D. Monderer and L. S. Shapley, “Potential games,” Games and Economic Behavior, vol. 14, no. 1, pp. 124 – 143, 1996.

[24] J. F. Nash, “Equilibrium points in n-person games,” Proceedings of the National Academy of Sciences, vol. 36, no. 1, pp. 48–49, 1950.

[25] E. Koutsoupias and C. Papadimitriou, “Worst-case equilibria,” in Proceedings of the 16th Annual Conference on Theoretical Aspects of Computer Science, ser. STACS’99. Berlin, Heidelberg: Springer-Verlag, 1999, pp. 404–413.

[26] M.-F. Balcan, B. Manthey, H. Röglin, and T. Roughgarden, “Analysis of Algorithms Beyond the Worst Case (Dagstuhl Seminar 14372),” Dagstuhl Reports, vol. 4, no. 9, pp. 30–49, 2015.

[27] S. Boyd and L. Vandenberghe, Convex optimization. Cambridge university press, 2004.