Optimal Utility Design of Greedy Algorithms in Resource Allocation Games

Rohit Konda †, Student Member, IEEE, Rahul Chandan ‡, Student Member, IEEE, David Grimsman ††, Member, IEEE, and Jason R. Marden †‡, Fellow, IEEE

Abstract—Designing fast, distributed algorithms for multiagent problems is vital for many novel application domains. Greedy algorithms have been shown in many multiagent contexts to be an efficient approach to arrive at good solutions quickly. In this work, we ask the following: Is there any way to improve the performance of greedy algorithms without sacrificing the linear run-time guarantees? For this, we take inspiration from incentive design in the game-theoretic literature. In this work, we consider a modified version of the greedy algorithm where agents do not optimize against the global objective. Instead, each agent is prescribed an auxiliary utility function (which may differ from the original objective function) in which it optimizes under. By designing the utility functions properly, we show in this work that the resulting performance guarantees of the greedy algorithm can increase significantly. We study this approach in the context of resource-allocation games, which are used to model a rich variety of engineering applications. Interestingly, the performance guarantees from the modified greedy algorithm can be significantly close to the best centralized performance guarantees. The main technical contribution of the article is the characterization of the performance guarantees through a linear program construction.

Index Terms—Algorithm design, distributed systems, game theory, optimization, resource allocation.

I. INTRODUCTION

THE analysis and control of multiagent systems has received a significant amount of attention in recent years, due to its tremendous potential for solving various important problems. Some pertinent examples where these systems have found success include wireless communication networks [2], UAV swarm task allocation [3], news subscription services [4], vaccinations during an epidemic [5], facility location [6], coordinating the charging of electric vehicles [7], and national defense [8], among others. The quintessential challenge in designing algorithms for these scenarios is to arrive at well-performing system-level behavior, as measured by some given global objective, that emerges in a distributed and scalable fashion. Therefore, the system designer is tasked to construct distributed algorithms that satisfy certain locality constraints while optimizing a given global objective.

One classic distributed design is the greedy algorithm, where the agents are ordered sequentially, and at each step of the execution, a single agent optimizes the global objective unlaterally given the previous agents’ decisions in the sequence. In general, the greedy algorithm is not guaranteed to find the globally optimal solution, but is a quick and elegant way to derive an approximately optimal solution. In fact, in certain well-structured domains, there may even be provable guarantees on the approximation ratios. For example, greedy algorithms are known to have good approximation guarantees in set covering problems [9], \( k \)-extendible problems [10], submodular maximization problems [11], etc. Instead of the standard implementation of the greedy algorithm, we consider a nonoblivious version, where each agent alternatively optimizes a given auxiliary utility function. While the utility functions may be misaligned with the original objective, they can be carefully designed so that poor solutions of the standard greedy algorithm are avoided [12], [13]. In this fashion, we can extend the greedy algorithm to ensure the resulting solutions have better worst-case quality with respect to the original objective.

In many applications, it may be preferable to implement the greedy algorithm over more sophisticated polynomial-time distributed algorithms. For example, there may be an extremely large number of agents in the multiagent scenario or the relevant situational parameters may be time-varying or there may be computational and run-time restrictions on the agents. In these instances, the linear execution time of the greedy algorithm is required. Thus, we extend the greedy algorithm to a nonoblivious version as a method to improve the corresponding performance guarantees while preserving the linear run-time of the algorithm.

Designing utility functions for agents is, in fact, a significant focus in the field of game theory, under the nomenclature of mechanism (or incentive) design. A vast literature in this field dedicates focus to designing the utility functions through incentives such that local solutions in the form of Nash equilibrium have desirable performance guarantees. This has been
done in a variety of contexts, including taxing structures in congestion games [14], mechanism design in auctions [15], and others [16], [17], [18]. Thus, in this article, we adopt ideas and tools from game theory when designing the utility functions for the nonoblivious version of the greedy algorithm. While taking inspiration from the game-theoretic literature, we shift focus from solutions of Nash equilibrium to the solutions of the nonoblivious greedy algorithm. As such, this work belongs to a larger research trend that aims to study game theoretic models beyond their respective equilibrium (see, for e.g., [19], [20], [21], [22], [23], [24], [25], [26], and [27]).

We utilize the connection between game theory and nonoblivious greedy algorithms to characterize and optimize the performance guarantees of the resulting solutions. We study the respective approximation guarantees from different utility function designs in the context of the well-studied class of resource allocation games. This class of games can model many relevant engineering applications [28], a sample of which is discussed in Section II. To the best of the authors’ knowledge, this is the first work in this context that improves the approximation guarantees of the greedy algorithm through a nonoblivious version. In this vein, the main results of this article are as follows. All proofs of the results are found in the Appendix.

1) Theorem 1 reformulates the characterization of efficiency guarantees of a given nonoblivious greedy algorithm as a linear program construction. While this linear program has an infinite number of constraints, a finite truncation of the constraint set gives tractable lower bounds on the performance guarantees.

2) We, then, adapt the linear program construction to resource allocation games with submodular and supermodular properties. In these instances, we can derive the optimal nonoblivious greedy algorithm, either through an optimization program in Theorem 2 in the submodular case or in a closed-form expression in Theorem 4 in the supermodular case.

3) If the set of submodular resource allocation games are parameterized through the notion of curvature, we can derive closed-form expressions of the optimal nonoblivious greedy algorithm and the resulting efficiency guarantees in Theorem 3. Furthermore, we show significant improvements over the standard greedy algorithm, shown in Fig. 1.

II. MODEL

This work considers multiagent scenarios in the form of resource allocation games [28]. Resource allocation games are characterized by a finite set of resources \( \mathcal{R} = \{r_1, \ldots, r_d\} \) that can be utilized by a set of agents \( \mathcal{I} = \{1, \ldots, n\} \). Each agent \( i \in \mathcal{I} \) has a finite action set \( \mathcal{A}_i \subseteq 2^\mathcal{R} \) representing the decisions available to each agent. The performance of a joint action \( a = (a_1, \ldots, a_n) \in \mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n \) comprised of every agents’ actions is evaluated by a system-level objective function \( W : \mathcal{A} \rightarrow \mathbb{R}_{>0} \) of the form

\[
W(a) = \sum_{r \in \bigcup_i a_i} w_r(|a_r|)
\]

where \( |a_r| = \{|i \in \mathcal{I} : r \in a_i\} \) is the number of agents that choose resource \( r \) in action profile \( a \), and the welfare rule \( w_r : \mathbb{N} \rightarrow \mathbb{R}_{>0} \) defines the resource-specific welfare determined by the utilization of \( r \) by \( |a_r| \) agents. The goal is for the agents to locally coordinate to a joint action that is close to the maximum welfare \( \max_{a \in \mathcal{A}} W(a) \), which is not known to the agents. In this form, we consider the greedy algorithm as an efficient method to do so. In the classical definition, the agents are ordered and each agent sequentially optimizes their action with respect to the global objective. The resulting joint action can be formally stated below. We use \( a^{gr}_r := \emptyset \) to denote a null action in which the agent selects no resources.

**Definition 1:** The joint action \( a^{gr} \) is the solution to the greedy algorithm if the following expression holds for all agents \( i \):

\[
a^{gr}_i \in \arg \max_{a_i \in \mathcal{A}_i} W (a^{gr}_{1}, \ldots, a^{gr}_{i-1}, a_i, a^{gr}_{i+1}, \ldots, a^{gr}_n).
\]

In other words, each agent selects the decision that maximizes the welfare given the decisions of the previous agents in the sequence. In this work, we extend the standard greedy algorithm by considering a nonoblivious version. In this nonoblivious algorithm, agents are still ordered sequentially, but they instead optimize their decisions with respect to some given utility functions \( U_i : \mathcal{A} \rightarrow \mathbb{R} \), where \( U_i \) can be designed to avoid poor solutions. The joint action that results from the nonoblivious algorithm can be defined similarly as before.

**Definition 2:** The joint action \( a^{no} \) is the solution to the nonoblivious greedy algorithm with utility functions \( \{U_i\}_{i \in \mathcal{I}} \)
if the following expression holds for all agents $i$:

$$a_i^{no} \in \arg \max_{a_i \in A_i} U_i \left( a_1^{no} , \ldots , a_i^{no} , a_{i-1} , a_{i+1} , \ldots a_n^{no} \right).$$

The central focus of this article is to understand how the choice of utility functions $U_i$ can impact the efficiency of the emergent solution $a^{no}$ of the nonoblivious algorithm. To this end, we define the following competitive ratio

$$\text{Eff}(G) = \frac{W(a^{no})}{\max_{a \in A} W(a)} \in [0, 1]$$

where a ratio closer to 1 implies the performance of the nonoblivious algorithm is closer to optimal. We use $G$ (formally defined later) to define a game with a certain welfare function and utility functions. As such, the objective of this work is to design the utility functions $U_i$ to maximize the competitive ratio of the respective nonoblivious algorithm.

Since this work is concerned with the design of distributed algorithms, we focus on utility functions in which the utility of agent $i$ is only locally dependent on the resources selected in $a_i$. Moreover, we restrict attention to utility functions of the form

$$U_i(a_i, a_{-i}) = \sum_{r \in A_i} u_r(|a_r|)$$

where the utility rule $u_r : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$ defines the resource-specific agent utility determined by $|a_r|$. We use $a_{-i} = (a_1, \ldots , a_{i-1}, a_{i+1}, \ldots a_n)$ to denote the joint action without the action of agent $i$. Now, the choice of utility rules $u_r$ influences the emergent guarantees of the nonoblivious greedy algorithm.

For the class of utility functions above, we will formally express a given resource allocation game by the tuple $G = \{\mathcal{I}, \mathcal{A}, \mathcal{R}, \{w_r\}_{r \in \mathcal{R}}, \{u_r\}_{r \in \mathcal{R}}\}$. In many distributed scenarios, a system designer may not have specific knowledge of the global parameters of the game, such as the number of agents $n$ or the joint action set $A$. To address this, we assume that the utility functions are only locally dependent on the resource types, or that $u_r$ is solely a function of $w_r$. To that end, let $W$ be the set of possible welfare rules that could be associated with any resource, i.e., $w_r \in W$ for all $r \in R$. A chosen utility function design can then be described as the mapping $\mathcal{U} : W \rightarrow \mathbb{R}_0^+$ corresponding to the utility rule $u_r = \mathcal{U}(w_r)$ for any resource $r \in R$.

We define the possible set of resource allocation games that can stem from a given $W$ and $\mathcal{U}$ as $\mathcal{G}_{W,\mathcal{U}}$, where a game $G \in \mathcal{G}_{W,\mathcal{U}}$ if and only if $w_r \in W$ and $u_r = \mathcal{U}(w_r)$ for all resources $r \in R$. As the specific game realization $G \in \mathcal{G}_{W,\mathcal{U}}$ is not known a priori, we extend the efficiency measure to be the worst-case guarantee

$$\text{Eff}(G) = \inf_{G \in \mathcal{G}_{W,\mathcal{U}}} \text{Eff}(G).$$

The directive of this article is to be able to quantify $\text{Eff}(G)$ from a given a welfare set $W$ and utility design $\mathcal{U}$ and derive the nonoblivious greedy algorithm that optimizes the respective efficiency guarantees. More formally, we would like to characterize

$$\text{Eff}_{no}(W) = \sup_{\mathcal{U} : W \rightarrow \mathbb{R}_{\geq 0}} \text{Eff}(\mathcal{G}_{W,\mathcal{U}})$$

for a given class of welfare rules $W$. Furthermore, we compare the guarantee $\text{Eff}_{no}(W)$ of the optimal nonoblivious algorithm with the guarantee $\text{Eff}_{\mathcal{U}}(W)$ of the standard greedy algorithm. The main results in Section III address these points in detail.

### A. Examples of Resource Allocation Games

We highlight a selection of applications that are well-modeled through resource allocation games below.

**Example 1 (Wireless Sensor Coverage):** Consider a group of $n$ sensors that can observe the region $R$ as in [29]. Each sensor has the ability to sense a subset of the region depending on its orientation, physical placement of the sensor, etc. The choice of these parameters constitute the action set $A_i$ for each agent, where a resource is monitored by the sensor if $r \in a_i$. As a whole, the set of sensors wish to arrive at a configuration, as dictated by a distributed process, that maximizes the likelihood of detecting an event. As such, the system welfare is

$$W(a) = \sum_{r \in \bigcup_{i \in \mathcal{I}} a_i} p_r \cdot \left(1 - (1 - p_d)^{|a_r|} \right)$$

where $p_r \in [0, 1]$ indicates the probability that the event will occur at $r$ and $p_d$ refers to the conditional probability that the event will be detected by a single sensor given that the sensor is monitoring $r$ and an event does indeed occur at $r$.

**Example 2 (Market Sharing Games):** In the setting considered in [30], the market sharing game is defined by a set of markets $R$ with $v_r$ being the value of each market $r \in R$. Each agent can select a portion of the markets in their action set $A_i$ and incurs a payoff of

$$U_i(a_i, a_{-i}) = \sum_{r \in A_i} \frac{v_r}{|a_r|} + \tau_r(|a_r|)$$

where the fraction of the market value $v_r$ is equally split among the agents that utilize market $r$ and $\tau_r(|a_r|)$ is a specified incentive mechanism (usually in the form of taxes). Individually, agents would like to maximize the usage of high-value markets, while minimizing overlap from other agents. As a whole, we would like to maximize the total market usage by agents, and so the respective global welfare function is defined as

$$W(a) = \sum_{r \in \bigcup_{i \in \mathcal{I}} a_i} v_r.$$

If there is a sequential selection of markets by agents via a schedule, we would like to maximize the resulting solution through manipulating the incentive mechanism $\tau_r$.

**Example 3 ($k$-Clustering):** Consider a classical dimensionality reduction problem of distilling a given data set into representative clusters, similar to [31]. Each data point $d_i$ has a set of possible representative clusters $A_i \subseteq R$ that it can join. The objective is to compute clusterings with maximum
overlap between data points. Greedy algorithms can provide quick solutions with respect to the following welfare function:

\[ W(a) = \sum_{r \in \mathcal{A}_a} (|a|_r)^2. \tag{11} \]

### III. MAIN RESULTS

Our main results address the performance of nonoblivious greedy algorithms in resource allocation games. Under this distributed algorithm design, each agent runs a local optimization to decide its action in a sequential fashion. Given a set of allowable welfare rules \( \mathcal{W} \) and a utility function design \( \mathcal{U} \), our first main result in Theorem 1 derives the efficiency guarantees of the respective nonoblivious algorithm through a linear program construction.

Let \( w_\ell \in \mathcal{W} \) and \( \mathcal{U}(w_\ell) = u_\ell \) for some index \( \ell \). We use the notation \( \bar{\mathcal{w}}_\ell(i) = w_\ell(i)/w_\ell(1) \) and \( \tilde{\mathcal{w}}_\ell(i) = u_\ell(i)/u_\ell(1) \) to simplify the presentation of the results. In addition, we make the mild assumption that \( u_\ell(1) = w_\ell(1) \) to normalize the utility rules.

**Theorem 1:** Let \( \mathcal{W} \) be the welfare set. Consider the nonoblivious algorithm with a utility function design \( \mathcal{U} \), where \( \mathcal{U}(w_\ell) = u_\ell \) for each \( w_\ell \in \mathcal{W} \). The resulting efficiency guarantee is \( \text{Eff}(\mathcal{G}_{\mathcal{W}, \mathcal{U}}) = \inf_{w_\ell \in \mathcal{W}} 1/\beta(w_\ell) \), where \( \beta(w_\ell) \in [1, \infty) \) is the solution to

\[
\beta(w_\ell) = \min \beta \quad \text{subject to:} \\
\beta \tilde{w}_\ell(y) \geq H \left( \sum_{i=1}^{y} \bar{\mathcal{w}}_\ell(i) - z \min_{1 \leq i \leq y+1} \bar{\mathcal{w}}_\ell(i) \right) + \bar{\mathcal{w}}_\ell(z) \\
\text{for all } z, y \in \mathbb{N} \text{ s.t. } z \geq 0 \text{ and } y \geq 1 \tag{12} \]

and \( H = \sup\{ \tilde{\mathcal{w}}_\ell(i)/i : i \in \mathbb{N}, w_\ell \in \mathcal{W} \} \).

**Remark 1:** Observe that the value \( \beta \tilde{w}_\ell \) is equal to the welfare rule \( \tilde{w}_\ell = a \cdot w_\ell \) for any \( a > 0 \) if the corresponding utility rule also satisfies \( u_\ell = a \cdot u_\ell \). For a finite collection of welfare rules, say \( \{u_1, \ldots, u_m\} \), consider the set \( \mathcal{W} = \{w : w = \sum_{j=1}^{m} a_j w_j : a_j \geq 0 \text{ for all } j\} \) that can be defined by the possible nonnegative linear combinations. In this instance, the quantity \( \inf_{w_\ell \in \mathcal{W}} 1/\beta(w_\ell) \) is then equal to \( \min\{1/\beta(w_1), \ldots, 1/\beta(w_m)\} \), which can be computed from a finite set of linear programs.

**Remark 2:** Note that if the number of agents \( n \) is known, the linear program in (12) provides a nontrivial lower bound for \( 1/\beta(w_\ell) \) when only including the constraints for \( z, y \leq n \). Thus, it is possible to derive lower bounds on the efficiency guarantees through a set of tractable optimization programs.

The above theorem sets forth a prescriptive process by which to characterize the efficiency guarantees of the nonoblivious algorithm through a linear program construction.3 This is done through a novel parameterization of the set of resource allocation games and careful elimination of the redundant constraints in the dual of resulting program. While directly solving the optimization in (12) requires keeping track of a countable number of constraints, this linear program construction provides valuable insights into the achievable efficiency guarantees. Under certain subclasses of welfare rules, the optimal utility function design that optimizes \( \text{Eff}(\mathcal{W}) \) can actually be derived in closed form. This is done in Sections III-A and III-B with regard to submodular and supermodular welfare rules.

### A. Submodular Resource Allocation Games

In this section, we will restrict attention to welfare rules that are submodular, or informally, welfare rules that admit a notion of decreasing marginal returns that are commonplace in many objectives relevant to engineered systems. Many well-studied applications including viral marketing, information gathering, image segmentation, and statistical summarization involve welfare objectives that are submodular (see [32] for a survey on application domains). Examples (1) and (2) involve welfare rules that are submodular. We formally define submodular welfare rules below.

**Definition 3 (Submodularity):** A welfare rule \( w \) is submodular if \( w \) is nondecreasing and concave in \( j \), or equivalently that \( w(j + 1) - w(j) \) is nonnegative and nonincreasing in \( j \).

Under the assumption of submodularity, we can simplify the linear program in Theorem 1.

**Corollary 1:** Let \( \mathcal{W} \) be a set of submodular welfare rules. Consider the nonoblivious algorithm with a utility function design \( \mathcal{U} \). The resulting efficiency guarantee is \( \text{Eff}(\mathcal{G}_{\mathcal{W}, \mathcal{U}}) = \inf_{w_\ell \in \mathcal{W}} 1/\beta(w_\ell) \), where \( \beta(w_\ell) \) is given by (12) with \( H = 1 \).

Furthermore, given a set of submodular welfare rules, we can derive the optimal nonoblivious algorithm, as well as its respective efficiency guarantees, through a corresponding linear program. The construction of the program is derived from the characterization result in Corollary 1 with the nontrivial fact that the optimal utility rules are nonincreasing in this domain.

**Theorem 2:** Consider the set \( \mathcal{W} \) where each \( w_\ell \in \mathcal{W} \) is a submodular welfare rule. The utility rules \( w^*_{\ell, u} \) of the optimal nonoblivious algorithm are given by the solutions to

\[
(w^*_{\ell, u}, \beta(w_\ell)) \in \arg \min_{\beta, u \in \mathbb{R}^{\mathbb{N}}_+} \beta \quad \text{subject to:} \\
\beta w_\ell(y) \geq \sum_{i=1}^{y} u(i) - u(z) + w_\ell(z) \quad \forall y, z \geq 1 \\
u(1) = w_\ell(1) \tag{13} \]

with a corresponding efficiency guarantee of \( \text{Eff}_{\text{opt}}(\mathcal{W}) = \inf_{w_\ell \in \mathcal{W}} 1/\beta(w_\ell) \).

**Remark 3:** Note that the optimization problem in (13) is intractable to solve directly. However, if we fix the number of agents to \( n \), and only consider the variables \( u \in \mathbb{R}_+^{\mathbb{N}} \) that are nonincreasing and constraints for \( 1 \leq y \) and \( z \leq n \), we can derive lower bounds on the optimal efficiency guarantees.

Thus, in the submodular setting, it is possible to derive characterizations of the optimal nonoblivious algorithm. While computing these characterizations is intractable in general, it is possible to compute the optimal nonoblivious algorithm in closed form for certain classes of welfare rules. In fact, this is possible if the submodular welfare rules are parameterized.

---

3 We note that the linear program in (12) is decoupled for each welfare rule \( w_\ell \).
through their curvature. Curvature is a classical parameterization used widely in submodular optimization problems (see [33] and [34]) that characterizes the rate of diminishing returns associated with a submodular welfare function. We note that any submodular welfare function has a curvature $C \in [0, 1]$. In our setting, curvature can be defined as follows.

**Definition 4 (Curvature):** A submodular welfare rule $w$ has a curvature of $C \in [0, 1]$ if $C = 1 - \lim_{n \to \infty} (w(n) + 1) - w(n)/w(1)$.

With this, we can arrive at a tight, closed-form characterization of the optimal performance guarantees, as shown below.

**Theorem 3:** Let the set $W$ comprise of all submodular welfare rules $w$ that have curvature of at most $C \in [0, 1]$. The efficiency guarantees of the optimal nonoblivious and standard greedy algorithms are defined by

$$\text{Eff}_{no}(W) = 1 - \frac{C}{2} \quad (14)$$

$$\text{Eff}_{gr}(W) = (1 + C)^{-1}. \quad (15)$$

Furthermore, the utility rules $u^o_w = U_{no}(w)$ for the optimal nonoblivious algorithm can be compactly expressed as

$$u^o_w(j) = \sum_{b \in \mathbb{N}} a_b u_b(j) \quad (16)$$

where $a_b \in \mathbb{R}_{\geq 0}$ and $u_b$ are defined in (33) and (35).

**Remark 4:** We note that the efficiency guarantees, in (15), of the standard greedy algorithm exactly matches the bound given for general submodular set functions [33].

In Theorem 3, we have characterized the efficiency guarantees of the optimal nonoblivious and the standard greedy algorithm in closed form. A visual comparison of the guarantees is depicted in Fig. 1. We also compare the efficiency guarantees to the best approximation guarantee $1 - C/e$ that is achievable by any polynomial time algorithm [35, Th. 2] in this setting. By only carefully designing the objectives that agents greedily optimize against, we see that there can be significant gains in the performance guarantees in submodular resource allocation games.

### B. Supermodular Resource Allocation Games

In this section, we consider welfare rules that are *supermodular*. Under this welfare structure, cooperative resource utilization results in a surplus of system welfare. Applications of supermodular games include clustering [see Example (3) for more details] and power allocation in networks [35]. A formal definition of supermodular welfare rules is as follows.

**Definition 5 (Supermodularity):** A welfare rule $w$ is supermodular if $w$ is nondecreasing and convex in $j$, or that $w(j + 1) - w(j)$ is nonnegative and nondecreasing in $j$.

Unlike in the submodular setting, the efficiency guarantees of the optimal nonoblivious and standard greedy algorithms can be characterized in closed form for supermodular welfare rules. This is done in the following theorem.

**Theorem 4:** Consider the set $W$ where each $w \in W$ is a supermodular welfare rule. The efficiency guarantees of the optimal nonoblivious and standard greedy algorithm are

$$\text{Eff}_{no}(W) = \text{Eff}_{gr}(W) = \inf_{w \in W} \lim_{n \to \infty} \frac{n}{w^o(n)}. \quad (17)$$

Furthermore, the utility rules $u^o_w = U_{no}(w)$ for the optimal nonoblivious algorithm is any rule that is nondecreasing and satisfies

$$\sum_{i=1}^j u^o_w(i)/w^o(j) \leq 1 \text{ for all } j \geq 1.$$

**Remark 5:** We can similarly define curvature for supermodular welfare rules, where $C = 1 - \lim_{n \to \infty} (w(n + 1) - w(n))/w(1)$. Under this definition, note that the efficiency guarantees in Theorem 4 can be equally stated as $\text{Eff}_{no}(W) = (1 - C)^{-1}$.

**Remark 6:** We remark the efficiency guarantees $\text{Eff}_{no}(W)$ of the optimal nonoblivious algorithm exactly matches the guarantees of Nash-seeking algorithms with optimal sharing rules [35, Th. 4]. Thus, greedy algorithms have similar guarantees to more complex algorithm designs.

We can consider the nonoblivious algorithm with a Shapley (or equal-shares) [36] utility function design for supermodular settings. Shapley utility functions are desirable due to their well-known budget balance property [28], where $\sum_a U_a(W) = W(a)$ for all joint actions $a \in A$. We observe that the Shapley utility rules (defined in this setting as $U^*(w) = u_{\text{shap}}$ with $u_{\text{shap}}(j) = w(j)/j$ for all $j \in \mathbb{N}$) satisfies the assumptions of Theorem 4, and thus, maximizes the possible efficiency guarantees. However, we note that the optimal utility rules are not unique, as the constant utility function design (defined as $U(w) = u_1$ with $u_1(j) = w(1)$ for all $j \in \mathbb{N}$) also satisfies the assumptions of Theorem 4. In addition, the standard greedy algorithm also has equivalent guarantees for supermodular welfare rules. However, the average case guarantees of different nonoblivious algorithms may be different; we leave it to future work to classify these algorithms based on their average behavior.

### IV. SIMULATED LOWER BOUND GUARANTEES

In this section, we compare the efficiency guarantees of nonoblivious algorithms with the standard greedy algorithm in Fig. 2. We do this by utilizing the results from Theorem 1 for a distribution of various welfare rules. Specifically, given a welfare rule $w$ and respective utility rule $u$ defined only on a finite number of entries, an efficiency guarantee $\text{Eff}(g_{\{w\},\{u\}})$ can be derived from Theorem 1 through considering only a finite number of constraints.

First, we randomly generate 5000 welfare rules $w \in \mathbb{R}^{15}$ defined for 15 entries. We do this by fixing $w(1) = 1$ and sampling the difference $w(j) - w(j-1)$ uniformly from $[0, 1]$ for all $2 \leq j \leq 15$ to recursively generate a normalized and monotone welfare rule. For each welfare rule, we derive the efficiency guarantee of the standard greedy algorithm by running the linear program in (12), with $H = \max_{i} \tilde{w}(i)/i$ and the marginal contribution utility rule (see Appendix C.1 for more details). In addition, for each welfare rule, 200 normalized utility rules are
randomly generated, where \( u(1) = 1 \) and \( u(j) \) is uniformly sampled from \([0, 1]\) for \( 2 \leq j \leq 15 \), and the efficiency guarantees for each utility rule are derived. The maximum calculated efficiency from this set is taken as a lower bound for the guarantees of the optimal nonoblivious algorithm. Accordingly, we generate a histogram of efficiency guarantees for the standard greedy algorithm and the optimal nonoblivious algorithm, shown in Fig. 2.

While the calculated efficiency results may not be tight for both the standard and nonoblivious greedy algorithms, this simulation highlights the potential gains from considering a nonoblivious design. As the mean guarantee of the standard greedy algorithm is 0.28 and the mean guarantee of the nonoblivious algorithm is 0.54, we see an improvement of 93% in the mean efficiency guarantee. We remark that since the nonoblivious algorithm has equivalent run-time as the standard greedy algorithm and the optimal nonoblivious algorithm, these gains are realizable just through manipulating the local objectives for agents.

V. Conclusion

Greedy algorithms provide fast and efficient solutions to a variety of multiagent problems. Inspired by the results in the game-theoretic literature, we focus on extending the performance guarantees of the greedy algorithm through a nonoblivious approach, where the objective functions of agents are carefully designed. We do this in the context of resource allocation games, which model a variety of applications. Our main result is on characterizing the efficiency guarantee given a class of resource allocation games. Utilizing these results, we derive expressions of the optimal nonoblivious algorithm in resource allocation games with the properties of submodularity or supermodularity. Surprisingly, well-designed nonoblivious algorithms can potentially increase the efficiency guarantees significantly over standard greedy algorithm. Future work may comprise of extending the results to other game models or consider average-case efficiency analysis.

APPENDIX

We iterate through the proofs of the main theorems of the article, as well as provide relevant technical discussion and lemmas in the appendix. Relevant code is found at [37]. The outline of the provided technical proofs are as follows.

A) First, we give a proof of Theorem 1, where given a welfare set \( \mathcal{W} \) and utility design \( \mathcal{U} \), a linear program is formulated to characterize \( \text{Eff}(G_{\mathcal{W}, \mathcal{U}}) \).

B) A proof is given for Corollary 1 and Theorem 2 to determine the optimal utility rules and efficiency guarantees in submodular settings.

C) A proof of Theorem 3 is provided, where the results of Theorem 2 are refined for a class of submodular welfare rules with a given curvature.

D) Given a welfare set \( \mathcal{W} \) that is now instead supermodular, we show the optimal efficiency guarantees in Theorem 4 as well as the guarantees of the greedy algorithm.

Notation: Given a set \( S \), \( |S| \) represents its cardinality and \( \mathbb{I}_e \) describes the corresponding indicator function. \( \mathbb{I}_e(c) = 1 \) if \( e \in S \), 0 otherwise. We denote the index of the \( j \)th component of a vector \( v \) with \( v_j \) or \( v(j) \) interchangeably. We use 1 to denote a vector of all ones and 0 to denote a vector of all zeros. We use the denotation \( w(0) = u(0) = 0 \) and \( \bar{w}_i(i) = w_i(i)/w_i(1) \) and \( \bar{a}_i(i) = u_i(i)/u_i(1) \).

A. Proof of Theorem 1

1) Linear Program Formulation of the Nonoblivious Algorithm: We first give a linear program that computes the efficiency \( \text{Eff}(G_{\mathcal{W}, \mathcal{U}}) \) that is based on a search for a worst-case game construction \( G \in G_{\mathcal{W}, \mathcal{U}} \) that achieves the worst efficiency ratio for the given nonoblivious algorithm. Here, \( G_{\mathcal{W}, \mathcal{U}} \) denotes the set of games with a fixed \( n \) number of agents, set of welfare rules \( \mathcal{W} \) and utility function design \( \mathcal{U} \). We also make the assumption that the welfare set \( \mathcal{W} \) is finite, but generalize beyond this assumption later in the proof. A comparable primal-dual approach was also explored in [17] and [25] for different settings.

First, we apply a key observation that for a game \( G \), truncating the action set of each agent \( i \) to \( A_i = \{a_i^1, a_i^2, a_i^\text{opt}\} \) does not affect the efficiency metric \( \text{Eff}(G) \). Here, \( a_i^\text{opt} \) is the null action that does not select any resources, \( a_i^\text{opt} \) is the action that agent \( i \) takes under the nonoblivious algorithm [i.e., the action satisfying (3)], and \( a_i^\text{opt} \) is the action that agent \( i \) plays in a joint action that optimizes the welfare \( a_i^\text{opt} = \arg\max_{a_i \in A} W(a) \). Therefore, we can restrict attention to the class of games \( G_{\mathcal{W}, \mathcal{U}} \subseteq G_{\mathcal{W}, \mathcal{U}} \), where agents only have these three actions available without loss of generality. Furthermore, scaling \( W \) uniformly does not affect the ratio \( \text{Eff}(G) = W(a^\text{opt})/W(a^\text{opt}) \), and we can assume that \( W(a^\text{opt}) = 1 \) without loss of generality. So we aim to find a

4Note that \( a_i^\text{opt} \) and \( a_i^\text{opt} \) may be the same action, but using separate denotations does not affect the game structure. In addition, if \( a_i^\text{opt} \) is not unique, then, the one that performs the worst with respect to \( W \) is selected.
The possible resource allocations are enumerated by the following product set:
\[
P = \prod_{i \in I} \{\emptyset, \{a^o_i\}, \{a^{opt}_i\}, \{a^o_i, a^{opt}_i\}\}
\]
where each resource is classified by the agent actions that can select it. Then, the respective vectors in \(\{0, 1\}^n\) can be defined.

\[
b^p_i = \begin{cases} 1 & \text{if } a^o_i \in p_i, \ 0 \text{ otherwise} \\ a^{opt}_i = \begin{cases} 1 & \text{if } a^{opt}_i \in p_i, \ 0 \text{ otherwise} \\ \end{cases}
\]

where \(p \in P\) describes a resource type. We define the norm of \(b^p\) to be \(|b^p| = \sum_{i \in I} b^p_i\) (similarly for \(|o^p| = \sum_{i \in I} o^p_i\) and denote the number of nonzero elements before index \(i\) as \(|b^p|_{<i} = \sum_{j < i} b^p_j\). With this, we describe the linear program in the following lemma.

**Lemma 1:** Consider the welfare set \(W = \{w_1, \ldots, w_m\}\). For \(n\) agents, the efficiency guarantee of the nonoblivious algorithm with the utility function design \(U\) is

\[
\text{Eff}(G^n_{W,U})^{-1} = \max_{(\beta_i)_{i \in I}, \beta} \beta \text{ subject to: } \\
\beta \bar{w}_\ell(|o^p|) \geq \bar{w}_\ell(|b^p|) + \sum_{i \in I} \beta_i ([|b^p_i - o^p_i|) \bar{w}_\ell(|b^p|_{<i} + 1] \\
\text{for all } p \in P \text{ and } 1 \leq \ell \leq m.
\]

**Proof:** First, we show the equivalence of the optimization program proposed in (18) and the primal linear program described below. We later show that the dual of this primal program is exactly the linear program in (21). Note that each decision variable \(\eta^p_\ell \in \mathbb{R}_{\geq 0}\) is a real nonnegative number.

\[
\text{Eff}(G^n_{\eta, U})^{-1} = \max_{(\eta^p_\ell)_{p \in P}, (\eta^p_\ell)_{p \in P}} \sum_{1 \leq \ell \leq m, p \in P} \bar{w}_\ell(|o^p|) \cdot \eta^p_\ell \text{ s.t. } \] (22)

\[
\sum_{1 \leq \ell \leq m, p \in P} \bar{w}_\ell(|b^p|) \cdot \eta^p_\ell \leq 1
\]

\[
\sum_{1 \leq \ell \leq m, p \in P} [([b^p_i - o^p_i] \bar{w}_\ell(|b^p|_{<i} + 1] \cdot \eta^p_\ell \geq 0 \ \forall i \in I.
\]

For the equivalence, we first define a vector label for each resource \(r\) as \(\ell_r(i) = \{a_i \in A_i : \text{ if } r \in a_i\}\). This vector describes in what actions is the resource selected by each agent \(i\), with \(\ell_r \in P\). Furthermore, we denote the specific partition of the resource set with \(|\ell_r^{\ell_r} = \{r \in R : \ell_r = p, w_r = w_\ell\}\). Now, we show that \(W(a^{opt})\) in (18) matches (22)

\[
W(a^{opt}) = \sum_{r \in R} w_r(|a^{opt}|_r) = \sum_{1 \leq \ell \leq m, p \in P} w_r(|a^{opt}|_r)
\]

where \(\eta^p_\ell = |\ell_r^{\ell_r} \cdot \eta^p_\ell(1)\). The first equality is from the definition of the welfare function. The second equality results from partitioning the resource set. The third equality occurs by the fact that \(|a^{opt}|_r = \sum_{j \in I} 1_{a^o_j}(r) = |o^p|\) if \(r \in \ell_r^{\ell_r}\). In addition, the value \(w_r(|o^p|)\) is constant for any \(r \in \ell_r^{\ell_r}\). A similar argument can be made about the welfare of the nonoblivious action \(W(a^{opt})\), so (19) matches (23) as well.

Now, we show the utility constraint in (20) matches the constraint in (24). For conciseness, let \(a_1^1 = (a^o_1, a^{opt}_1, a^o_2)\) and \(a_2^2 = (a_1^1, a_2^{opt}, a^o_3)\). The utility difference can be written as

\[
U_i(a_1^1) - U_i(a_2^2) = \sum_{r \in R} u_r(|a_1^1|_r) - \sum_{r \in R} u_r(|a_2^2|_r)
\]

\[
= \sum_{1 \leq \ell \leq m, p \in P} \left(\sum_{r \in R} 1_{a^o_2}(r) u_r(|a_1^1|_r) - \sum_{r \in R} 1_{a^{opt}_2}(r) u_r(|a_2^2|_r)\right)
\]

\[
= \sum_{1 \leq \ell \leq m, p \in P} \left(\sum_{r \in R} 1_{a^o_2}(r) u_r(|a_1^1|_r) - \sum_{r \in R} 1_{a^{opt}_2}(r) u_r(|a_2^2|_r)\right)
\]

\[
= \sum_{1 \leq \ell \leq m, p \in P} \left(\sum_{r \in R} 1_{a^o_2}(r) u_r(|a_1^1|_r) - \sum_{r \in R} 1_{a^{opt}_2}(r) u_r(|a_2^2|_r)\right)
\]

The first equality is from the definitions of the utility functions. The second and third equalities come from rewriting the sum using indicator functions and partitioning the resource set along \(P\). The fourth equality is a result of three facts: that \(1_{a^o_2}(r) = b^p_1\); that \(1_{a^{opt}_2}(r) = o^p_1\); and that \(|a_1^1|_r = \sum_{j \leq i \leq 2} 1_{a^o_2}(r) + 1 = |b^p_1|_{<i} + 1 \) if \(r \in a^{opt}_2\) (similarly for \(|a_2^2|_r\)). The fifth equality comes from sliding out the relevant terms of the first sum and using the assumption that \(u_r(1) = w_r(1)\).

We assume that \(\eta^p_\ell \geq 0\) to ensure a well-defined game parameterization. Observe that in the primal program in (22), we have relaxed \(\eta^p_\ell \in \mathbb{R}_{\geq 0}\) to be any nonnegative real number with \(\eta^p_\ell\) denoting the relative fraction of resources with a specific resource type. We use this relaxation to normalize \(W(a^{opt}) = 1\) and this relaxation is done without loss of generality, since we can scale up the values \((\eta^p_\ell)_{p \in P}\) [from the solution arguments of (22)] uniformly and round to derive the resource set for a corresponding valid game construction \(G\) that achieves an
efficiency ratio \( \text{Eff}(G) \) that is arbitrarily close to the solution of the primal program.

We now verify that the dual of the primal program in (22) matches the linear program defined in (21). Note that primal program in (22) can be concisely written as

\[
\max_{\eta} \; c^T \eta \quad \text{subject to:}
\]

\[
K \eta = 1 \\
L \eta \\ I_m \quad \eta \geq 0
\]

where \( \eta \) is the vector of \( \{ \eta_p \}_{p \in \mathcal{P}} \) corresponding to the identity matrix of dimension \( m \cdot 4^n \times m \cdot 4^n \), and \( c, K, \) and \( L \) are the compactly written vectors in (22), (23), and (24), respectively. Writing the dual linear program gives

\[
\max_{\lambda \geq 0, \; \xi \geq 0, \; \beta} \; -\beta \quad \text{subject to:}
\]

\[
K^T \beta - [L^T, I_m^T] \begin{bmatrix} \lambda \\ \xi \end{bmatrix} - c_\ell = 0 \quad \forall 1 \leq \ell \leq m
\]

where \( c = (c_1, \ldots, c_m)^T \) is associated with each \( 1 \leq \ell \leq m \) (likewise for \( K \) and \( L \)). Observe that the constraint set \( K^T \beta - [L^T, I_m^T] \begin{bmatrix} \lambda \\ \xi \end{bmatrix} - c_\ell = 0 \) is equivalently written as \( K^T \beta - L^T \lambda - c_\ell = \xi \) and \( K^T \beta - L^T \lambda - c_\ell \geq 0 \). Substituting back \( c_\ell, K_\ell, \) and \( L_\ell \) results in the constraint

\[
\beta \bar{w}_\ell(\|b^p\|) \geq \bar{w}_\ell(\|a^p\|) + \sum_{i \in I} \lambda_i (\|b^p_i - a^p_i\|) \bar{w}_\ell(\|b^p_i\|_{c_i}) + 1]
\]

which matches the constraint outlined in (21).

2) Continuing the Proof of Theorem 1: The dual program in (21) provides a solution for \( \text{Eff}(G_{D, k, \mathcal{U}})^{-1} = \beta^* \) for a fixed \( n \) and finite \( \mathcal{W} \). However, the constraint set is exponential in the number of agents. Thus, in this section, we remove redundant constraints to arrive at a more tractable linear program. We first show the solution is upper bounded by \( \beta^* \leq \beta \) for any \( n \), where \( \beta = \max_{1 \leq \ell \leq m} \beta(w_j) \) and \( \beta(w_j) \) is the solution to the program in (22).

Let \( n \) be the number of agents. Without loss of generality, we assume that \( w_\ell(1) = w_\ell(1) = 1 \) for \( 1 \leq \ell \leq m \). For a given \( p \in \mathcal{P} \), we denote \( y_p = \|b^p\| \) and \( z_p = \|a^p\| \) for ease of notation. In addition, to convey which indices the resource type \( p \) are nonzero in and in which order, we define vectors for \( a^{\text{no}} \) and \( O^p \) for \( a^{\text{op}} \). Formally, \( B^p : \{1, \ldots, y_p\} \to \{1, \ldots, n\} \) and \( O^p : \{1, \ldots, z_p\} \to \{1, \ldots, n\} \) with

\[
B^p(j) = i \text{ if } b^p_i = 1 \text{ and } |b^p|_{\leq i} = j \\
O^p(j) = i \text{ if } a^p_i = 1 \text{ and } |a^p|_{\leq i} = j.
\]

Considering the dual program in (21), we add the constraint that \( \lambda_i = H = \sup_{i \in \mathbb{N}, 1 \leq i \leq m} w_\ell(i) / i \) explicitly. Since we shrink the feasible region, the optimal solution to (21) potentially increases. We verify that the resulting feasible region is nonempty. Consider the constraints according to \( p \), such that \( b^p = 0 \). The corresponding dual constraint takes the form

\[
0 \geq w_\ell(z_p) - \sum_{j=1}^{z_p} \lambda_{O^p(j)} u_\ell(1)
\]

Simplifying the expression gives \( \sum_{j=1}^{z_p} \lambda_{O^p(j)} \geq w_\ell(z_p) \), which is always satisfied if \( \lambda_i = H \) for all \( i \). If the constraints according to \( p \) are such that \( b^p \neq 0 \), then the term \( \beta w_\ell(y) \) is present and strictly positive in the inequality (21) and \( \beta \) can be taken as high as needed to satisfy the constraint. Therefore, the feasible region is nonempty.

For any \( p \in \mathcal{P} \) such that \( b^p \neq 0 \), we can simplify the dual constraint in (21), for each \( \ell \), to

\[
\beta w_\ell(y_p) \geq w_\ell(z_p) + \sum_{i=1}^{y_p} H u_\ell(i) - \sum_{i \in I} H o^p u_\ell(\|b^p\|_{c_i} + 1).
\]

Furthermore, for any \( p \in \mathcal{P} \), we observe that

\[
\sum_{i \in I} o^p u_\ell(\|b^p\|_{c_i} + 1) = z_p \min_{1 \leq i \leq y_p + 1} u_\ell(i).
\]

Thus, for any \( p \in \mathcal{P} \), we can replace the corresponding dual constraint with a more binding constraint

\[
\beta w_\ell(y_p) \geq w_\ell(z_p) + \sum_{i=1}^{y_p} H u_\ell(i) - \sum_{i \in I} H z_p \min_{1 \leq i \leq y_p + 1} u_\ell(i)
\]

for some \( 0 \leq z_p \leq n \) and \( 1 \leq y_p \leq n \). Therefore, re-placing the dual constraints gives an upper bound for \( \beta^* \leq \beta \).

Since \( \beta \) is the only variable in the optimization problem, we can decouple the constraints for each \( \ell \) and limit the number of agents \( n \to \infty \) to arrive at the program in (12).

Now, we show that the solution is lower bounded by \( \beta^* \geq \beta \), where \( \beta^* \) and \( \beta \) are defined as before. We show that when we remove dual constraints, we arrive at the set of linear programs in (12). Since the feasible region expands, the optimal solution potentially decreases. Let the set of agents be \( \mathcal{I} = \mathbb{N} \) and \( j^p_p = \arg \min_{1 \leq i \leq y_p + 1} u_\ell(i) \). We remove all the dual constraints barring the constraints that correspond to \( p \in \mathcal{P} \) with either (a) \( y_p = 0 \) or \( z_p = z^p \), or (b) \( y_p > 0 \) and \( B^p(j^p_p - 1) < O^p(1) \) and \( O^p(z^p) < B^p(j^p_p) \). The first program refers to all resource types where \( a^{\text{no}} \) is never selected but \( a^{\text{op}} \) is by \( z^p \) agents. The second program refers to all resource types where the indices of the agents selecting \( a^{\text{op}} \) are between the agents with index \( B^p(j^p_p - 1) \) and \( B^p(j^p_p) \). Assume property (a). Then, the corresponding dual constraint in (21) can be written as

\[
0 \geq w_\ell(z^p) - \sum_{j=1}^{z^p} \lambda_{O^p(j)} u_\ell(1)
\]

for any resource type \( p \in \mathcal{P} \) that satisfies property (a) and for all \( \ell \). Therefore, for any \( j \in \mathbb{N} \), except for at most \( z^p - 1 \) (with \( z^p = \max \{z^p\} \) values, observe that \( \lambda_j \geq H \) must hold.

Now, assume property (b). With respect to a resource type \( p \in \mathcal{P} \) that satisfies property (b), we observe that \( u_\ell(\|b^p\|_{c_i} + 1) = u_\ell(j^p_p) \) for any agent with index \( i = O^p(j) \) for some \( j \). Therefore, under the two previous observations, we can rewrite the relaxed dual program as
\[
\begin{align*}
\min_{\beta \geq 0} \beta \quad \text{subject to:} \\
\beta u_t(y_p) \geq \sum_{j=1}^{y_p} \lambda_{BP}(j) u_t(j) - \sum_{j=1}^{z_p} \lambda_{OP}(j) u_t(j^p) + w_t(z_p) \\
\end{align*}
\]
for all \( p \in \mathcal{P}' \) and \( \ell \\
\lambda_i \geq H \quad \text{for all } i \in \mathbb{N} \text{ but at most } z^* - 1 \text{ values} \tag{25}
\]
where \( \mathcal{P}' = \{ p \in \mathcal{P} : p \text{ satisfies property (b)} \} \). Observe that we recover the proposed program given in (12) if we assume that the optimal dual variable is \( \lambda_i = H \) for all \( i \in \mathbb{N} \). To show this claim, we confirm that the binding constraint for \( \beta \) in (25) is larger when considering a different sequence of \( \lambda_i \neq H \). In other words, for a given \( y \geq 1 \) and \( z \geq 0 \), we show that for the resulting dual variables
\[
\beta_{\lambda} := \max_{p \in \mathcal{P}'} \left\{ \frac{1}{w_t(y_p)} \left( \sum_{j=1}^{y_p} \lambda_{BP}(j) u_t(j) - \sum_{j=1}^{z_p} \lambda_{OP}(j) u_t(j^p) \right) \right\}
\geq \frac{H}{w_t(y)} \left( \sum_{j=1}^{y} u_t(j) - \sum_{j=1}^{z} u_t(j^p) \right) := \beta_{y,z}. \tag{26}
\]
For any \( \lambda \neq H \), consider two cases where either \( \lambda \) is a divergent sequence, or it is bounded above. In the first case, since \( \lambda \) must satisfy \( \lambda_j \geq 0 \) for all \( j \in \mathbb{N} \), the limit \( \lim_{j \to \infty} \lambda_j = 0 \). If \( u_t(j) = 0 \) for all \( j \), note that \( \beta_{y,z} \) is zero for any \( y \geq 1 \) and \( z \geq 0 \). Since, \( \beta_{\lambda} \) must also be greater than zero, the inequity in (26) holds in this case. If \( u_t(J) > 0 \) for some \( J \in \mathbb{N} \), consider a constraint with \( p \) such that \( y_p > J \) and \( z_p = 0 \). For any \( M > 0 \), we can choose \( B^p \), such that \( \lambda_{BP}(j) > M \) for all \( 1 \leq j \leq y_p \). Thus, \( \beta_{\lambda} \geq 1/w_t(y_p) \sum_{j=1}^{y_p} M u_t(j) \). Since \( M \) is arbitrary, \( \beta_{\lambda} = \infty \geq \beta_{y,z} \) for any \( y \geq 1 \) and \( z \geq 0 \) as well.

In the second case, since \( \lambda \) is also bounded below by \( H \), for all but a finite set of values of \( \lambda \), there exists a converging subsequence \( \lambda_{ss} \) that converges to a value \( V \geq H \) by the Bolzano–Weierstrass theorem. Let \( M_a = \max_{1 \leq j \leq y+1} u_t(i) \), \( x = \max(y,z) \), and \( \varepsilon > 0 \). Since \( \lambda_{ss} \) converges, there exists a \( J \in \mathbb{N} \) such that for any \( j \geq J \), \( \lambda_{ss}(j) - V | \leq \varepsilon/2 M_a \).

For a given \( y \) and \( z \), consider any constraint with \( p \in \mathcal{P}' \), such that \( y_p = y \) and \( z_p = z \). In addition, \( B^p \) and \( O^p \) can be chosen to ensure that \( |\lambda_{BP}(j) - V| \leq \varepsilon/2 M_a \) and \( |\lambda_{OP}(j) - V| \leq \varepsilon/2 M_a \) for all \( j \). Therefore
\[
\beta_{\lambda} \geq \frac{1}{w_t(y_p)} \left( \sum_{j=1}^{y_p} \lambda_{BP}(j) u_t(i) - \sum_{j=1}^{z_p} \lambda_{OP}(j) u_t(j^p) \right) \geq \frac{V}{w_t(y)} \left( \sum_{j=1}^{y} u_t(i) - \sum_{j=1}^{z} u_t(j^p) \right) - \frac{\varepsilon}{2} - \frac{\varepsilon}{2} \geq \beta_{y,z} - \varepsilon.
\]
Since \( \varepsilon \) is arbitrary, we have that \( \beta_{\lambda} \geq \beta_{y,z} \) for any \( y \) and \( z \) and we show the claim. Therefore, the proposed program is an upper bound for any \( n \) and we have shown the equality \( \beta^* = \beta \) for any \( n \).

Note that the welfare rule set \( \mathcal{W} \) was assumed to be finite for the previous arguments. Now, we extend to more general sets of welfare rules. As the worst-case efficiency is defined as \( \text{Eff}(G_{WU}) = \inf_{G \in G_{WU}} \text{Eff}(G) \), for a given sequence \( \varepsilon_j \to 0 \), there always exist a game \( G_j \in G_{WU} \), such that \( \text{Eff}(G_j) \leq \text{Eff}(G_{WU}) + \varepsilon_j \). Take a sequence of \( \varepsilon_j \to 0 \). Define a finite welfare set for each step as \( W_j := \bigcup_{1 \leq k \leq j} W_r : r \in R_k \) described as the union of the welfare rules for each game in the sequence. For each step, we have that \( \text{Eff}(G_{WU}) \leq \beta \leq \text{Eff}(G_j) \leq \text{Eff}(G_{WU}) + \varepsilon_j \), where \( \beta \) is the solution derived from the set of linear programs in (12) for the welfare set \( W_j \). Taking \( j \to \infty \) gives the result.

\[\Box\]

B. Proof of Corollary 1 and Theorem 2

1) Proof of Corollary 1: We directly apply Theorem 1 to derive the efficiency guarantees for submodular welfare rules. Note that if \( w_t \) is submodular, \( \bar{w}_t(i) \leq i \) for all \( i \). Thus, we can substitute \( H = \sup_{x \in \mathcal{X}} \bar{w}_t(i)/i = \bar{w}_t(1)/1 \).

2) Proof of Theorem 2: We simply refer to \( \bar{w}_t \) as \( w \) in the following discussion. If the utility rule \( u \) is assumed to be nonincreasing, we will show that we recover the linear program in (13). If \( u \) is nonincreasing, then, \( \min_{1 \leq i \leq y+1} u(i) = u(y+1) \). In addition, \( w(1) - 1 \cdot u(y+1) \geq w(0) - 0 \cdot u(y+1) = 0 \) for any \( y \geq 1 \), so \( z = 0 \) is a nonbinding constraint. We lastly note that the values \( \{u(i)\}_{i \in \mathcal{X}} \) can be established as decision variables for the program in (12) to produce the linear program in (13), rewritten below.

\[
\begin{align*}
(\beta^*, u^*) & \in \arg \min_{\beta, \{u(i)\}_{i \in \mathcal{X}}} \beta \quad \text{subject to:} \\
\beta w(y) & \geq \sum_{i=1}^{y} u(i) - zu(y+1) + w(z) \quad \forall y, z \geq 1 \\
u(1) & = 1
\end{align*}
\tag{27}
\]
where \( \beta^* \) is a tight characterization of the efficiency guarantee only if the resulting optimal utility rule \( u^* \) is nonincreasing and a lower bound if not. We now verify that the optimal utility rule \( u^* \) is indeed nonincreasing for this simplified program. First, rearranging the terms in the constraint in (27) gives that for any \( y \geq 1 \),

\[
\begin{align*}
u^*(y+1) & \geq \sup_{z \geq 1} \left( \frac{1}{z} \left( \sum_{i=1}^{y} u^*(i) + w(z) - \beta^* w(y) \right) \right).
\end{align*}
\tag{28}
\]
We verify \( u^*(y+1) \) is well-defined. Note that since \( u^* \) is optimal, the efficiency bound \( \beta^* < \infty \) is nontrivial (as the standard greedy algorithm guarantees an efficiency guarantee greater than 1/2 [18]). Then, by recursion and the fact that \( w(z)/z \leq 1 \) for all \( z \), there exists a solution for \( u^*(y+1) \), such that (28) holds with equality and the resulting value is finite for all \( y \geq 1 \). In addition, \( u^*(y) \) must be nonnegative for all \( y \geq 1 \), since limiting \( z \to \infty \) in (28) gives that \( u(y+1) \geq 0 \).

Now, we show that the solution \( u^* \) is nonincreasing. Suppose for contradiction that for some \( y \geq 1 \), that \( u^*(y) < u^*(y+1) \). Let \( z_{y+1} \in \arg \max_{z \geq 1} w(z) - zu(y+1) \) be the number that achieves the maximum.

We verify that \( z_{y+1} \) is well-defined. Suppose for contradiction that \( w(z) - zu^*(y+1) \) is always increasing in \( z \), so \( z_{y+1} \).
is not well defined. Since \( \beta^* < \infty \), the limit \( \lim_{z \to \infty} w(z) - zu^*(y + 1) \) must converge and, therefore, \( u^*(y + 1) \) must be equal to \( Q = \lim_{z \to \infty} \Delta w(z) \), where we denote \( \Delta w(z) = w(z) - w(z - 1) \) for conciseness. From the original contradiction assumption, then, \( u^*(y) < u^*(y + 1) = Q \). Then, taking the constraint in (27), with \( y - 1 \) and \( z \to \infty \) gives \( \beta w(y - 1) \geq \lim_{z \to \infty} w(z) - zu^*(y) \geq \infty \), which is a contradiction.

Now, substituting \( y_{j+1} \) into (28) for \( y \) and \( y + 1 \) produces the following expressions:

\[
\begin{align*}
    u^*(y + 1) &= \frac{1}{y_{j+1}} \left( \sum_{i=1}^{y} u^*(i) + w(z_{j+1}) - \beta^* w(y) \right) \\
    u^*(y) &\geq \frac{1}{y_{j+1}} \left( \sum_{i=1}^{y-1} u^*(i) + w(z_{j+1}) - \beta^* w(y - 1) \right).
\end{align*}
\]

Inputting these expressions into the assumption \( u^*(y) < u^*(y + 1) \) reduces to the inequality \( u(y) < \beta^* \Delta w(y) \). Similarly, for some \( j \geq 1 \), substituting \( y_{j+1} \) into (27) for \( y + j \) and \( y + j + 1 \) gives

\[
\begin{align*}
    u^*(y + j + 1) &\geq \frac{1}{y_{j+1}} \left( \sum_{i=1}^{y+j} u^*(i) + w(z_{j+1}) - \beta^* w(y + j) \right) \\
    u^*(y + j) &= \frac{1}{y_{j+1}} \left( \sum_{i=1}^{y+j-1} u^*(i) + w(z_{j+1}) - \beta^* w(y + j - 1) \right).
\end{align*}
\]

Thus, by substituting the second expression into first, the following inequality holds:

\[
\begin{align*}
    u^*(y + j + 1) &\geq u^*(y + j) + \frac{u^*(y + j) - \beta^* \Delta w(y + j)}{y_{j+1}}.\tag{29}
\end{align*}
\]

We show, by induction, that the following expression holds for any \( j \geq 1 \):

\[
\frac{u^*(y + j) - \beta^* \Delta w(y + j)}{y_{j+1}} \geq \frac{u^*(y + 1) - \beta^* \Delta w(y + 1)}{y_{j+1}} > 0.\tag{30}
\]

The base case holds for \( j = 1 \), since

\[
u^*(y + 1) - \beta^* \Delta w(y + 1) > u^*(y) - \beta^* \Delta w(y) > 0.
\]

This comes from the assumption that \( u^*(y + 1) > u^*(y) \), \( \Delta w(y + 1) \leq \Delta w(y) \) by submodularity of \( w \), and that \( u^*(y) - \beta^* \Delta w(y) > 0 \) from the previous argument. For the inductive case for \( J \geq 2 \), assume that the inequality holds for all \( j < J \). Then, by applying the induction assumption to (29), and subsequently to the definition of \( z_{y+j} \), we have that

\[
u^*(y + J) > u^*(y + J - 1) > \ldots > u^*(y + 1) \geq \frac{u^*(y + j) - \beta^* \Delta w(y + j)}{y_{j+1}} > \frac{u^*(y + j - 1) - \beta^* \Delta w(y + j - 1)}{y_{j+1}} > \cdots \geq \frac{u^*(y + 1) - \beta^* \Delta w(y + 1)}{y_{j+1}}.
\]

Therefore, the statement in (30) holds due to the aforementioned inequalities and the fact that \( \Delta w(y + J) \leq \Delta w(y + 1) \) due to submodularity of \( w \). Therefore, (30) holds and we have that \( u^*(y + j + 1) \geq u^*(y + j) + D \), where \( D = u^*(y + 1) - \beta^* \Delta w(y + 1)z_{y+1} > 0 \). Following this, \( u^*(y + j + 1) \geq u^*(y + 1) + D(j - 1) \).

Now, consider the constraint in (27) where \( y \to \infty \) and \( z = 0 \). Since \( w(y) \leq y \)

\[
\beta^* \geq \lim_{y \to \infty} \frac{1}{y} \sum_{i=1}^{y} u^*(i) \geq \infty\tag{31}
\]

where the last inequality results from the fact that \( u^*(y) \sim y \) is of linear order by the previous argument. Since \( \beta^* \) must be finite, contradiction ensues and the solution \( u^* \) must be nonincreasing and the efficiency guarantees are tight for the linear program. \( \Square \)

C. Proof of Theorem 3

1) Proof of Efficiency for the Standard Greedy: We verify the equality in (15). We first note that the solution to the standard greedy algorithm in Definition 1 is equivalent to the solution to the nonoblivious algorithm with a utility function design (also known as marginal contribution [28]) \( U_i(a_i, a_{-i}) = W(a_i, a_{-i}) - W(a_i', a_{-i}) \) for all \( i \) and \( a \). This utility function design can also be alternatively defined through the utility rules \( u^mc_\ell = u_\ell(w_\ell) \) that satisfy \( u^mc_\ell(j) = w_\ell(j) - w_\ell(j - 1) \) for all \( j \). Now, we can use Corollary 1 to characterize the efficiency guarantee of the standard greedy algorithm.

Consider the set \( \mathcal{W} \) of submodular welfare rules that have curvature of at most \( C \) and assume \( w_\ell(1) = 1 \) without loss of generality. The utility rule \( u^mc_\ell \) must be nonincreasing, and the constraints in (12) can be rewritten as

\[
\beta w_\ell(j) \geq \sum_{i=1}^{y} u^mc_\ell(i) - zw^mc_\ell(y + 1) + w_\ell(z) \tag{32}
\]

for any \( y \geq 1 \) and \( z \geq 0 \). We claim the binding constraint is when \( z = y \). Fixing \( y \), the only terms that depend on \( z \) is \( -zw^mc_\ell(y + 1) + w_\ell(z) \). Examining the difference between terms from \( z + 1 \) against \( z \) gives

\[
\begin{align*}
    w_\ell(z + 1) - (z + 1)w^mc_\ell(y + 1) - w_\ell(z) + zw^mc_\ell(y + 1) \\
    &= w_\ell(z + 1) - w_\ell(z) - w^mc_\ell(y + 1) \\
    &= u^mc_\ell(z + 1) - w^mc_\ell(y + 1),
\end{align*}
\]

since \( u^mc_\ell \) is nonincreasing, note that \( u^mc_\ell(z + 1) - u^mc_\ell(y + 1) \) is greater than 0 if \( z < y \) and less than 0 if \( z > y \). Therefore, the tightest constraint is when \( z = y \). Now, we simplify the solution for \( \beta(w_\ell) \) in (12) under the assumption that \( z = y \) as

\[
\beta(w_\ell) = \max_{y \geq 1} \left\{ \frac{1}{w_\ell(y)} \left( \sum_{j=1}^{y} u^mc_\ell(j) - yu^mc_\ell(y + 1) + w_\ell(y) \right) \right\} \]

\[
\max_{y \geq 1} \left\{ 2 - \frac{y}{w_\ell(y)} u^mc_\ell(y + 1) \right\}
\]

in which we have used the identity \( \sum_{j=1}^{y} u^mc_\ell(j) = \sum_{j=1}^{y} w_\ell(j) - w_\ell(j - 1) = w_\ell(y) \). Since \( w_\ell \) is submodular, \( h/\ell w_\ell(j) \geq 1 \) for any \( j \in \mathbb{N} \), and because \( w_\ell \) has at most curvature of \( C \), \( u^mc_\ell(j) \geq 1 - C \) for any \( j \in \mathbb{N} \) as well. Therefore, the solution is upper bounded by \( \beta(w_\ell) \leq 1 + C \) and since \( w \) was chosen arbitrarily from \( \mathcal{W} \), the resulting efficiency guarantee
is \( \text{Eff}(G_{W, U_{mc}}) = \inf_{t} 1/\beta(w_{l}) \geq (1 + C)^{-1} \). This efficiency guarantee is actually tight if we consider the \( b \)-covering welfare rule \( w^{b} \) with curvature \( C \), as in (34). Observe that under the \( b \)-covering welfare, the maximum is \( \max_{y \geq 1} y/w^{b}(y)w_{mc}(y + 1) = 1 - C \) at \( y = b \). Therefore, \( \text{Eff}(G_{W, U_{mc}}) = (1 + C)^{-1} \) with equality and we show the claim.

2) Proof of Efficiency for the Nonoblivious: We verify the equality in (14) and structure of the optimal utility rules. Given a curvature \( C \), let \( \mathcal{W} \) be the set of welfare rules that have curvature of at most \( C \). From [34, Lemma 2], we know there exists a basis set of welfare rules, such that for any \( w \in \mathcal{W} \), we can come up with a decomposition \( w = \sum_{b \in \mathbb{N}} \alpha^{b}w^{b} \), with

\[
\alpha^{b} = \frac{(2w(b) - w(b) - 1 - w(b + 1))}{C} \quad \text{and} \quad w^{b}(j) = \begin{cases} j, & \text{if } 0 \leq j \leq b \\ b + (1 - C) \cdot (j - b), & \text{if } j > b. \end{cases}
\]

(33)

We refer to these welfare rules as \( b \)-covering welfare rules. We note that for any \( b \in \mathbb{N} \), the welfare rule \( w^{b} \) has a curvature of \( C \). We consider a linear utility function design \( \mathcal{U}_{lin}(w_{l} = \sum_{b \in \mathbb{N}} \alpha^{b}w^{b}) = \sum_{b \in \mathbb{N}} \alpha^{b}w^{b} \). Note that the constraint in (12) is satisfied for any linear combination of \( w^{b} \) and \( w^{b} \), so we only need to confirm optimality of \( w^{b} \) for each. For each welfare rule \( w^{b} \), we claim that the corresponding optimal utility rule from running the program in (13) is

\[
u^{b}(j) = \begin{cases} (1 - \beta^{b})(\frac{b+1}{b})^{j-1} + \beta^{b}, & \text{if } j \leq b + 1 \\ (1 - C)\beta^{b}, & \text{if } j \geq b + 1 \end{cases}
\]

(35)

where \( \beta^{b} = (\frac{b+1}{b})^{b}/(\frac{b+1}{b})^{b} - C \) is the resulting optimal efficiency. Taking the minimum across \( b \), we have that \( \min_{b \in \mathbb{N}} \frac{1}{\tau^{b}} = 1 - C/2 \) for \( b = 1 \). Therefore, using Theorem 2, the optimal efficiency guarantee is \( \text{Eff}_{\text{opt}}(\mathcal{W}) = 1 - C/2 \).

Now, we verify that \( u^{b} \) and \( \beta^{b} \) are indeed the optimal solutions. We first remove all constraints in (13) apart from the ones that satisfy \( z = b \) for any \( y \geq 1 \). This results in a lower bound for \( \beta^{b} \) that we claim later to be tight. Rearranging the terms in the constraint in (27) gives that for any \( y \geq 1 \), the optimal solution satisfies

\[
u^{*}(y + 1) = \sup_{z \geq 1} \left( \frac{1}{z} \left( \sum_{i=1}^{y} u^{*}(i) + w(z) - \beta^{*}w(y) \right) \right).
\]

(36)

Substituting in for \( w \) and the binding constraint \( z = b \), the recursive equation for \( u^{b} \) is then

\[
u^{b}(1) = 1
\]

\[
u^{b}(j + 1) = \frac{1}{b} \sum_{i=1}^{j} u^{b}(i) + 1 - \frac{1}{b} \beta^{*}w^{b}(j)
\]

for some optimal \( \beta^{*} \geq 1 \). To solve for the closed-form expression for \( w^{b} \), a corresponding linear, time-invariant, discrete time system is constructed as follows:

\[
x_{1}(t + 1) = x_{1}(t) + x_{2}(t)
\]

\[
x_{2}(t + 1) = \frac{1}{b}(x_{1}(t) + x_{2}(t)) + s(t)
\]

\[
s(t) = 1 - \frac{1}{b} \beta^{*}w^{b}(t).
\]

For the initial condition \( (x_{1}(1), x_{2}(1)) = (0, 1) \), the corresponding solution \( x_{2}(t) = \sum_{t=0}^{\infty} \beta^{t}x_{1}(t) \) results in the closed-form solution for \( x_{2}(t) \) as

\[
x_{2}(1) = 1
\]

\[
x_{2}(t) = \frac{1}{b}B^{t-2} \left( 1 + \sum_{t=1}^{\infty} B^{-\tau}(1 - \beta^{*}) \right)
\]

\[
+ (1 - \beta^{*}w^{b}(t - 1)) \quad t > 1
\]

(37)

where \( B = b + 1/b \). Simplifying the expression for \( x_{2}(t) \) for \( t - 1 > b \) and substituting \( w^{b}(t) = (1 - C)t + C \min (t, b) \) results in the following:

\[
x_{2}(t) = \frac{1}{b}B^{t-2} \left( 1 + \sum_{t=1}^{\infty} B^{-\tau}(1 - \beta^{*}) \right)
\]

\[
+ (1 - \beta^{*}(t - 1 - C(t-1) + C))B).
\]

Thus, the above expression is the closed-form solution for \( u^{b} \) when \( j - 1 > b \). We have already shown that the optimal utility rule \( w^{b} \) must be nonincreasing in the proof of Theorem 2. This is only possible when \( \beta^{b} \geq B^{b}/B^{b} - C \). Therefore, the optimal solution must be \( \beta^{*} = \beta^{b} = B^{b}/B^{b} - C \). Substituting for \( \beta^{*} \) in the expression in (37) and simplifying results in the closed-form expression in (35) for \( u^{b} \). It can be seen that \( u^{b} \) defined in (35) is indeed nonincreasing. We lastly verify that the binding constraint for \( u^{b} \) is indeed when \( z = b \) for any \( y \geq 1 \) and so \( \beta^{b} \) is tight. In (13), we examine the terms \( w^{b}(z) - wz^{b}(y + 1) \) for any \( y \geq 1 \). Note that \( 1 = w^{b}(z) - w^{b}(z - 1) \geq w(z)(y + 1) \) when \( z \leq b \) and \( (1 - C) = w^{b}(z) - w^{b}(z - 1) \leq w(y)(y + 1) \) when \( z \geq b \) for any \( y \). Thus, the maximum \( \max_{y} w^{b}(z) - wz^{b}(y + 1) \) occurs when \( z = b \), and we have shown the claim.

D. Proof of Theorem 4

We first show that \( \text{Eff}_{\text{opt}}(\mathcal{W}) \leq \min_{1 \leq t \leq m} \lim_{n \to \infty} n/\tilde{w}_{1}(n) \) for a finite supermodular welfare rule set.\(^5\) We do this through a game construction, depicted in Fig. 3. Let \( w^{*} = \arg \min_{1 \leq t \leq m} n/\tilde{w}_{1}(n) \) be the welfare rule that attains the minimum. Let the game \( G \) have \( n \) agents with agent \( i \) having the action set \( A_{i} = \{ a_{i}^{0}, a_{i}^{1}, a_{i}^{2}, a_{i}^{3} \} \). There are \( n + 1 \) resources, which are all endowed with the welfare rule \( w_{r}^{*} = w^{*} \) for all \( r \in \mathcal{R} \), with agent \( i \) either selecting \( a_{i}^{0} = r_{i+1} \) or \( a_{i}^{3} =\)

\(^5\)We assume that the welfare set \( W \) is finite for ease of presentation, and it is straightforward to extend to more general sets of welfare rules.
{r_1}. Under any utility rule u, each agent i is indifferent to choosing a^opt_i or a^opt_j if no other agents j ≠ i have selected r_1 through a^opt_j. Thus, a^opt is a possible solution with a welfare of W(a^opt) = n · w^*(1). The welfare of the optimal allocation a^opt is W(a^opt) = w^*(n). Therefore, we have that Eff_no(W) ≤ Eff(G) = min_{1≤r≤m,n} n/\bar{w}_r(n) for any n and this is increasing in n so we have the claim.

Now, we show that for a utility design U^, such that the utility rule u_\ell = u_{\ell}(w) is nondecreasing and satisfies \sum_{i=1}^{n} u_\ell(i)/w(i) ≤ 1 for every j and \ell, the one-round efficiency is lower bounded by Eff(G^\ell,W,\ell) ≥ \min_{1≤r≤m,n} n/\bar{w}_r(n) for all n. To do this, we can use a modified version of the linear program in (12) for n agents, in which Eff(G^\ell,W,\ell) ≥ \min_{1≤r≤m} 1/\beta_\ell, where \beta_\ell \in \mathbb{R} is the solution to

\beta_\ell = \min \beta \quad \text{subject to:}

\beta \bar{w}_\ell(y) ≥ H \sum_{i=1}^{y} u_\ell(i) - z \min_{1≤r≤y+1} \bar{u}_\ell(i) + \bar{w}_\ell(z)

for all 0 ≤ z ≤ n and 1 ≤ y ≤ n

where the linear program is a lower bound, since we consider tighter constraints that allow y and z to range from 1 to n. Since w_\ell is supermodular, H = max_y \bar{w}_\ell(n)/n and assuming u_\ell is nondecreasing, \min_{1≤r≤y+1} \bar{u}_\ell(i) = \bar{u}_\ell(1) = 1. Thus, we can simplify the constraint as

\beta ≥ \left( H \sum_{i=1}^{y} \bar{u}_\ell(i) - Hz + \bar{w}_\ell(z) \right) / \bar{w}_\ell(y) \quad \text{(38)}

With this, we observe that \bar{w}_\ell(z) = Hz is convex in z. So the binding constraint for z occurs at either the end point z = 0 or z = n. Observe that max_{y} \bar{w}_\ell(n) = Hn, \bar{w}_\ell(0) = 0 and the terms can be canceled out. In addition, max_{y} \sum_{i=1}^{y} \bar{u}_\ell(i) = 1 occurs at the binding constraint y = 1, by assumption that \sum_{i=1}^{y} u_\ell(i)/w(i) ≤ 1 for all 1 ≤ j ≤ n. Therefore, \beta_\ell = H = \max_y \bar{w}_\ell(n)/n for all \ell under the binding constraint of y = 1 and z = 0 and we indeed have that Eff_no(W) ≥ \min_{1≤r≤m,n} lim_{n→∞} n/\bar{w}_r(n).

Note that the marginal contribution utility rule (see Appendix C.1) satisfies the assumptions of optimality in Theorem 4 as \sum_{i=1}^{n} u^\text{mc}_{j,i} = w_\ell(j) for any j and u^\text{mc} is nondecreasing for supermodular welfare rules. Thus, the standard greedy algorithm inherits the same efficiency guarantee Eff_{\ell}(W) = Eff_no(W).
V. S. Mirrokni and A. Vetta, “Convergence issues in competitive games,” in Approximation, Randomization, and Combinatorial Optimization Algorithms and Techniques, Berlin, Germany: Springer, 2004, pp. 183–194.

I. Caragiannis, M. Flammini, C. Kaklamanis, P. Kanellopoulos, and L. Moscardelli, “ Tight bounds for selfish and greedy load balancing,” in Proc. Int. Conf. Autom. Lang. Program., 2006, pp. 311–322.

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

C.-F. Huang and Y.-C. Tseng, “The coverage problem in a wireless sensor network,” Mobile Netw. Appl., vol. 10, no. 4, pp. 519–528, 2005.

B. Awerbuch, Y. Azar, A. Epstein, V. S. Mirrokni, and A. Skopalik, “Fast convergence to nearly optimal solutions in potential games,” in Proc. 9th ACM Conf. Electron. Commerce, 2008, pp. 264–273.

M. Narasimhan, N. Jocić, and J. A. Bilmes, “Q-clustering,” in Proc. Adv. Neural Inf. Process. Syst., 2005, pp. 979–986.

J. Chaitin, “Submodular function maximization,” Tractability, vol. 3, pp. 71–104, 2014.

M. Conforti and G. Cornuésjols, “Submodular set functions, matroids and the greedy algorithm: Tight worst-case bounds and some generalizations of the Rado-Edmonds theorem,” Discrete Appl. Math., vol. 7, no. 3, pp. 251–274, 1984.

R. Chandan, D. Paccagnan, and J. R. Marden, “Tractable mechanisms for computing near-optimal utility functions,” in Proc. 20th Int. Conf. Auton. Agents MultiAgent Syst., 2021, pp. 306–313.

D. Paccagnan and J. R. Marden, “Utility design for distributed resource allocation—Part II: Applications to submodular, covering, and supermodular problems,” IEEE Trans. Autom. Control, vol. 67, no. 2, pp. 618–632, Feb. 2022.

L. S. Shapley, “A value for n-person games,” in Contributions to the Theory of Games II, Princeton, NJ, USA: Princeton Univ. Press, 1953, pp. 307–317.

R. Konda, “kround,” 2022. [Online]. Available: https://github.com/rohit-konda/kround.git

**Rohit Konda** (Student Member, IEEE) received the B.S. degree in biomedical engineering in 2018 and the M.S. degree in electrical and computer engineering in 2019, from Georgia Tech, Atlanta, GA, USA. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, Santa Barbara, CA, USA.

He is currently a Regent’s Fellow with the University of California, Santa Barbara. His research interests include distributed optimization and game-theoretic applications in multiagent systems.

**Rahul Chandan** (Student Member, IEEE) received the B.A.Sc. degree in engineering science and electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2017, and the Ph.D. degree from the Electrical and Computer Engineering Department, University of California, Santa Barbara, CA, USA.

His research interests include the application of game theoretic and classical control methods to the analysis and control of multiagent systems.

**David Grimsman** (Member, IEEE) received the B.S. degree in electrical and computer engineering and the M.S. degree in computer science from Brigham Young University, Provo, UT, USA, in 2006 and in 2016, respectively, and the Ph.D. degree, under the supervision of Jason R. Marden and Joao P. Hespanha, from the Electrical and Computer Engineering Department, University of California, Santa Barbara, CA, USA, in 2021.

He was a Heritage Scholar with Brigham Young University, with a focus on signals and systems. He was a Trainer and an IT Manager with BrainStorm, Inc., for several years. He is currently an Assistant Professor with the Department of Computer Science, Brigham Young University. His research interests include multiagent systems, game theory, distributed optimization, network science, linear systems theory, and security of cyberphysical systems.

**Jason R. Marden** (Fellow, IEEE) received the B.S. and Ph.D. degrees in mechanical engineering, under the supervision of Jeff S. Shamma, from the University of California, Los Angeles (UCLA), Los Angeles, CA, USA, in 2001 and 2007, respectively.

After graduating from UCLA, he was a Junior Fellow with the Social and Information Sciences Laboratory, California Institute of Technology until 2010, when he joined the University of Colorado. He is currently a Full Professor with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA, USA. His research focuses on game theoretic methods for the control of distributed multiagent systems.

Dr. Marden was the recipient of the NSF Career Award in 2014, the Outstanding Graduating Ph.D. degree award, the ONR Young Investigator Award in 2015, the AFOSR Young Investigator Award in 2012, the American Automatic Control Council Donald P. Eckman Award in 2012, and the SIAG/CST Best SICON Paper Prize in 2015.