Cache me if you can: Capacitated Selfish Replication Games *

Ragavendran Gopalakrishnan[1], Dimitrios Kanoulas[2], Naga Naresh Karuturi[3], C. Pandu Rangan[4], Rajmohan Rajaraman[2], and Ravi Sundaram[2]

[1] California Institute of Technology, USA ragad3@caltech.edu
[2] Northeastern University, USA {dkanou, rraj, koods}@ccs.neu.edu
[3] Google India naganareshk@google.com
[4] IIT Madras, India rangan@iitm.ernet.in

Abstract. Motivated by peer-to-peer (P2P) networks and content delivery applications, we study Capacitated Selfish Replication (CSR) games, which involve nodes on a network making strategic choices regarding the content to replicate in their caches. Selfish replication games were introduced in [6], who analyzed the uncapacitated case leaving the capacitated version as an open direction.

In this work, we study pure Nash equilibria of CSR games with an emphasis on hierarchical networks, which have been extensively used to model communication costs of content delivery and P2P systems. The best result from previous work on CSR games for hierarchical networks [19, 23] is the existence of a Nash equilibrium for a (slight generalization of a) 1-level hierarchy when the utility function is based on the sum of the costs of accessing the replicated objects in the network. Our main result is an exact polynomial-time algorithm for finding a Nash Equilibrium in any hierarchical network using a new technique which we term "fictional players". We show that this technique extends to a general framework of natural preference orders, orders that are entirely arbitrary except for two constraints - "Nearer is better" and "Independence of irrelevant alternatives". This axiomatic treatment captures a vast class of utility functions and even allows for nodes to simultaneously have utility functions of completely different functional forms.

Using our axiomatic framework, we next study CSR games on arbitrary networks and delineate the boundary between intractability and effective computability in terms of the network structure, object preferences, and number of objects. In addition to hierarchical networks, we show the existence of equilibria for general undirected networks when either object preferences are binary or there are two objects. For general CSR games, however, we show that it is NP-hard to determine whether equilibria exist. We also show that the existence of equilibria in strongly connected networks with two objects and binary object preferences can be solved in polynomial time via a reduction to the well-studied even-cycle problem.

* Partially supported by a gift from NU alumnus Madhav Anand and NSF grants CCF-0635119 and CNS-0915985.

1 Introduction

Consider a P2P movie sharing service where you need to decide which movies to store locally, given your limited disk space, and which to obtain from your friends. Note that your decisions affect those of your friends, who in turn take actions that affect you. A natural question arises: what is the prognosis for you and your network of friends in terms of the stability of your movie collections and the satisfaction you will derive from them? Similarly, in the brave new wireless world of 4G you will not only be a consumer of different apps, you (your personal communications and computing device) will also be a provider of apps to others around you. And the question arises: could this lead to a situation of endless churn (in terms of what apps to store) or could there be an equilibrium?

In this paper, we study Capacitated Selfish Replication (CSR) Games, which provide an abstraction of the above scenarios. These are games in which the strategic agents, or players, are nodes in a network. The nodes have object preferences as well as bounded storage space – caches – in which they can store copies of the content. Each node cooperates with other nodes by serving their requests to access objects stored in its cache. However, the set of objects that a node chooses to store in its cache is entirely based on its own utility function and where objects of interest have been stored in the network.

Such a game-theoretic framework was first introduced in [6], which analyzed pure Nash equilibria in a setting with storage costs but no cache capacities, and left the capacitated version as an open research direction. Recent work on CSR games has focused on *hierarchical networks*, which are extensively used to model the communication costs of content delivery and P2P systems. (For instance, see [14] that uses the ultrametric model for content delivery networks and the work of of [20, 15, 16, 26] on cooperative caching in hierarchical networks.) The best result from previous work on CSR games for hierarchical networks [19, 23] is the existence of a Nash equilibrium for (a slight generalization of) a one-level hierarchical network using the sum utility function, i.e., when the utility of each node is based on a weighted sum of the cost of accessing the objects.

1.1 Our results

This paper studies the existence and computability of Nash equilibria for several variants of CSR games, with a particular focus on hierarchical networks. As with earlier studies [6, 19, 23, 1], we focus on the case where all pieces of content have the same size; note that otherwise even computing the best response of a player (node) is a generalization of the well-known knapsack problem and is NP-hard.

- Our main result is a polynomial-time algorithm for finding a Nash equilibrium for CSR games in any hierarchical network, thus resolving the question left open by [18, 23]. Our algorithm, presented in Section 3, is based on a new technique that we call the method of "fictional players" where we introduce and eliminate fictional players iteratively in a controlled fashion, maintaining a Nash equilibrium at each step, until the end when we have the desired equilibrium for the entire network (without any fictional players).

The above result is presented specifically in the context of the sum utility function to elucidate the technique of fictional players. We then abstract the central requirements for our proof technique and develop a general axiomatic framework to extend our results to a large class of utility functions.

– We present, in Section 4, a general framework for CSR games involving utility preference relations and node preference orders. Rather than specifying a numerical utility assigned by each node to each placement of objects, we only require that the preference order each node has on object placements satisfy two natural constraints of Monotonicity (or "Nearer is better") and Consistency (or "Independence of irrelevant alternatives"). This axiomatic treatment captures a vast class of utility functions and even allows for nodes to simultaneously have utilities of completely different functional forms.
– We extend our result for hierarchical networks to the broader class of utilities allowed by the axiomatic framework, and then study general CSR games obtained by considering different network structures (directed or undirected) and different forms of object preferences (binary or general). We delineate the boundary between intractability and effective computability of equilibria in terms of the network structure, object preferences, and the total number of objects. These results, presented in Sections 5 and 6, are summarized in Table 1. Notable results include: (1) the existence of equilibria for undirected networks with two objects using the technique of fictional players, (2) the existence of equilibria for undirected networks when object preferences are binary, and (3) the equivalence of finding equilibria in CSR games with two objects and binary object preferences to the even-cycle problem [24].

Object preferences and count	**Undirected networks**	**Directed networks**
Binary, two objects	Yes, in P (5)	No, in P (6)
Binary, three or more objects	Yes, in PLS (5)	No, NP-complete (6)
General, two objects	Yes, in P (5)	No, NP-complete (6)
General, three or more objects	No, NP-complete (6) Hierarchical: Yes, in P (3)	No, NP-complete (6)

Table 1. Existence and computability of equilibria in CSR games. Each cell (other than in the first row or the first column) first indicates whether equilibria always exist in the particular sub-class of CSR games. If equilibria always exist, then the cell next indicates the complexity of determining an equilibrium; otherwise, it indicates the complexity of determining whether equilibria exist for a given instance. The relevant subsection is given in parentheses.

1.2 Related work

In the last decade there has been a tremendous flowering of research at the intersection of game theory and computer science [21]. In a seminal paper [22] Papadimitriou laid the groundwork for algorithmic game theory by introducing syntactically defined subclasses of FNP with complete problems, PPAD being a notable such subclass. Subsequent work has identified a number of important problems in algorithmic game theory that are complete for PPAD [7, 5] or related complexity classes such as PLS [13].

Selfish caching games were introduced in [6] who considered the uncapacitated case where nodes could store more pieces of content by paying for the additional storage. We believe that limits on cache-capacity model an important real-world restriction and hence our focus on the capacitated version which was left as an open direction by [6]. Special cases of the integral version of CSR games have been studied. In [19], Nash equilibria were shown to exist for when nodes are equidistant from one another and a special server holds all objects. [23] slightly extends [19] to the case where special servers for different objects are at different distances. Our results generalize and completely subsume all these prior cases of CSR games. The Market sharing games defined by [11] also consider caches with capacity, but are of a very special kind; unlike CSR games, market sharing games are a special case of congestion games. In this work we focus primarily on equilibria and our general axiomatic framework has the flavor of similar frameworks from the theory of social choice [2, 21]; in this sense, we deviate from prior work [9, 8] that is focused on the price of anarchy [17].

There has been considerable research on capacitated caching, viewed as an optimization problem. Various centralized and distributed algorithms have been presented for different networks in [1, 3, 20, 15, 27].

2 A basic model for CSR games

We consider a network consisting of a set V of nodes labeled 1 through $n = |V|$ sharing a collection O of unit-size objects. For any i and j in V, let d_{ij} denote the cost incurred at i for accessing an object at j; we refer to d as the access cost function. We say that j is node i's *nearest* node in a set S of nodes if j is in S and $d_{ij} \leq d_{ik}$ for all k in S. We say that the given network is *undirected* if d is symmetric; that is, if $d_{ij} = d_{ji}$ for all i, j in V. We call an undirected network *hierarchical* if the access cost function forms an ultrametric; that is, if $d_{ik} \leq \max\{d_{ij}, d_{jk}\}$ for all $i, j, k \in V$.

Each node i has a cache to store a certain number of objects. The placement at a node i is simply the set of objects stored at i. The strategy set of a given node is the set of all feasible placements at the node. A *global placement* is any tuple $(P_i : i \in V)$, where $P_i \subseteq$ O represents a feasible placement at node i. For convenience, we use P_{-i} to denote the collection $(P_j : j \in V \setminus \{i\})$, thus often using $P = (P_i, P_{-i})$ to refer to a global placement. We also assume that V includes a (server) node that has the capacity to store all objects. This ensures that at least one copy of every object is present in the system; this assumption can be made without loss of generality since we can set the access cost of every node to this server to be arbitrarily large.

CSR **Games.** In our game-theoretic model, each node attaches a utility to each global placement. We assume that each node i has a weight $r_i(\alpha)$ for each object α representing the rate at which i accesses α. We define the *sum utility function*

$U_s(i)$ as follows: $U_s(i)(P) = -\sum_{\alpha \in \mathrm{O}} r_i(\alpha) \cdot d_{i\sigma_i(P,\alpha)}$, where $\sigma_i(P, \alpha)$ is i's nearest node holding α in P.

A CSR game is a tuple $(V, \mathrm{O}, d, \{r_i\})$. Our focus is on *pure Nash equilibria* (henceforth, simply *equilibria*) of the CSR games we define. An equilibrium for a CSR game instance is a global placement P such that for each $i \in V$ there is no placement Q_i such that $U_s(i)(P) > U_s(i)(Q)$.

Unit cache capacity. In this paper, we assume that all objects are of identical size. Under this model, we can assume without loss of generality that each node's cache holds exactly one object (see [12]).

3 Hierarchical networks

In this section, we give a polynomial-time construction of equilibria for CSR games on hierarchical networks. Any hierarchical network can be represented by a tree T whose set of leaves is the node set V and every internal node v has a label $\ell(v)$ such that (a) if v is an ancestor[5] of w in T, then $\ell(v) \geq \ell(w)$, and (b) for any i, j in V, d_{ij} is given by $\ell(\mathrm{lca}(i, j))$, where $\mathrm{lca}(i, j)$ denotes the least common ancestor of nodes i and j [14, 15].

Fictional players. In order to present our algorithm, we introduce the notion of a *fictional player*. For an object α, a *fictional α-player* is a new node that stores α in any equilibrium; for any fictional α-player ℓ, $r_\ell(\alpha)$ is 1 and $r_\ell(\beta)$ is 0 for any $\beta \neq \alpha$. Each fictional player is introduced as a leaf in the current hierarchy; the exact locations in the hierarchy are determined by our algorithm. The access cost function is naturally extended to the fictional players using the hierarchy and the labels of the internal nodes. In the following, we use "node" to refer to both the elements of V and fictional players.

A preference relation. The hierarchical network and the weights that nodes have for different objects induce, for each node i, a natural preorder $\sqsupseteq_i$ among elements of $\mathrm{O} \times A_i$, where A_i is the set of proper ancestors of i in T. Specifically, we define $(\alpha, v) \sqsupseteq_i (\beta, w)$ whenever $r_i(\alpha) \cdot \ell(v) > r_i(\beta) \cdot \ell(w)$. We can now express the best response of any player directly in terms of these preference relations. We define $\mu_i(P) = (\alpha, v)$ where $P_i = \{\alpha\}$ and v is $\mathrm{lca}(i, \sigma_i(P_{-i}, \alpha))$, where $\sigma_i(P_{-i}, \alpha)$ denotes i's nearest node in the set of nodes holding α in P_{-i}.

Lemma 1. *A best response P_i of a node i for a placement P_{-i} of $V \setminus \{i\}$ is $\{\alpha\}$ where α maximizes $(\gamma, lca(i, \sigma_i(P_{-i}, \gamma)))$, over all objects γ, according to $\sqsupseteq_i$.*

Proof. $U_s(i)(P) = -\sum_{\gamma \neq \alpha} r_i(\gamma) \ell(\mathrm{lca}(i, \sigma_i(P_{-i}, \gamma)))$, for a given placement P with $P_i = \{\alpha\}$. This can be rewritten as $-(\sum_{\gamma \in \mathrm{O}} r_i(\gamma) \ell(\mathrm{lca}(i, \sigma_i(P_{-i}, \gamma)))) + r_i(\alpha) \cdot \ell(\mathrm{lca}(i, \sigma_i(P_{-i}, \alpha)))$. Thus, $\{\alpha\}$ is a best response to P_{-i} if and only if α maximizes $r_i(\gamma) \cdot \ell(\mathrm{lca}(i, \sigma_i(P_{-i}, \gamma))$ over all objects γ. The desired claim follows from the definition of $\sqsupseteq_i$.

[5] We adopt the convention that each node is both descendant and ancestor of itself.

The algorithm. We introduce several fictional players at the start of the algorithm. We maintain the invariant that the current global placement is an equilibrium in the current hierarchy. As the algorithm proceeds, the set of fictional players and their locations change as we remove existing fictional players or add new ones. On termination, there are no fictional players leaving us with a desired equilibrium. Let W_t and P^t denote the set of fictional players and equilibrium, respectively, at the start of step t of the algorithm.

Initialization. We add, for each object α and for each internal node v of T, a fictional α-player as a leaf child of v; this constitutes the set W_0. The initial equilibrium P^0 is defined as follows: for each fictional α-player i, we have $P_i^0 = \{\alpha\}$; each node i in V plays its best response. Clearly, each fictional player is in equilibrium, by definition. Furthermore, for every α, every i in V has a sibling fictional α-player. Thus, the best response of every i in V is independent of the placement of nodes in $V \setminus \{i\}$, implying that P^0 is an equilibrium.

Step t of algorithm. Fix an equilibrium P^t for the node set $V \cup W_t$. If W_t is empty, then we are done. Otherwise, select a node j in W_t. Let $P_j^t = \{\alpha\}$, and let $\mu_j(P^t) = (\alpha, v)$. Let S denote the set of all nodes $i \in V$ such that $(\alpha, v) \sqsupset_i \mu_i(P^t)$. We now describe how to compute a new set of fictional players W_{t+1} and a new global placement P^{t+1} such that P^{t+1} is an equilibrium for $V \cup W_{t+1}$. We consider two cases.

- S is empty: Remove the fictional player j from W_t and the hierarchy, and leave the placement in the remaining nodes as before. Thus $W_{t+1} = W_t - \{j\}$ and P^{t+1} is the same as P^t except that P_j^{t+1} is no longer defined.
- S is nonempty: Select a node i in S such that $\mathrm{lca}(i, j)$ is lowest among all nodes in S. Let $P_i^t = \{\beta\}$. We set $P_i^{t+1} = \{\alpha\}$, remove the fictional α-player j from W_t, and add a new fictional β-player ℓ as a leaf sibling of i in T; i.e., $P_\ell^{t+1} = \{\beta\}$. For every other node j, set $P_j^{t+1} = P_j^t$. Finally, set $W_{t+1} = (W_t \cup \{k\}) \setminus \{j\}$.

Lemma 2. *For step t of the algorithm, if P^t is an equilibrium for $V \cup W_t$, then the following statements hold.*

1. *For every node k in $V \cup W_{t+1}$, P_k^{t+1} is a best response to P_{-k}^{t+1}.*
2. *For every node k in $V \cup W_{t+1}$, $\mu_k(P^{t+1}) \sqsupseteq_k \mu_k(P^t)$.*
3. *We have $|W_{t+1}| \le |W_t|$. Furthermore, either $|W_{t+1}| < |W_t|$ or there exists a node i in V such that $\mu_i(P^{t+1}) \sqsupset_i \mu_i(P^t)$.*

Proof. Let α, v, S, i, and j be as defined in step t of the algorithm above (see illustration in [12]). We first establish statements 1 and 2 of the lemma. Let k be any node in $V \cup W_{t+1}$. Consider first the case where $\mathrm{lca}(k, j)$ is an ancestor of v (i.e., k is not in the subtree rooted at the child u of v that contains j). For any object γ, we have $\sigma_k(P_{-k}^{t+1}, \gamma) = \sigma_k(P_{-k}^t, \gamma)$ and $P_k^{t+1} = P_k^t$. It thus follows that $\mu_k(P^{t+1}) = \mu_k(P^t)$, implying statement 2 for k. Since P^t is in equilibrium, statement 1 also holds for k.

We next establish statements 1 and 2 for any node k where $\mathrm{lca}(k, j)$ is a proper descendant of v (i.e., k is in the subtree rooted at the child u of v that contains j). We consider two cases. The first case is where S is empty. In this case, the fictional α-player j is removed; thus j is not in W_{t+1}. Furthermore, there is no copy of α in the subtree rooted at u. Since no object other than α is created or removed in this case, we have $\sigma_k(P^{t+1}_{-k}, \gamma) = \sigma_k(P^t_{-k}, \gamma)$ for $\gamma \neq \alpha$. We also have $\mathrm{lca}(k, \sigma_k(P^{t+1}_{-k}, \alpha)) = v$ and $\mu_k(P^{t+1}) = \mu_k(P_t)$, the latter establishing statement 2 for k. Since S is empty, $\mu_k(P^t) \sqsupseteq_k (\alpha, v)$. It follows from Lemma 1 and the fact that P^t_k is in equilibrium that P^{t+1}_k is a best response against P^{t+1}_{-k}, establishing statement 1 for k.

The second case is where S is not empty. Let i be as defined above, i.e., i is a node in S such that $\mathrm{lca}(i, j)$ is lowest among all nodes in S. Let x denote $\mathrm{lca}(i, j)$. Let P^t_i be equal to $\{\beta\}$, where $\beta \neq \alpha$. By the algorithm, we have $P^{t+1}_k = \{\alpha\}$. Let $k \neq i$ be a node in the subtree rooted at u. For any $\gamma \neq \alpha$, $\sigma_k(P^{t+1}_{-k}, \gamma) = \sigma_k(P^{t+1}_{-k}, \gamma)$. Since $P^{t+1}_k = P^t_k \neq \{\alpha\}$, we have $\mu_k(P^{t+1}) = \mu_k(P^t)$, establishing statement 2 for k. For node i, we have $\mu_i(P^{t+1}) = (\alpha, v) \sqsupset_i \mu_i(P^t)$, establishing statement 2 for i.

It remains to establish statement 1 for any node k in the subtree rooted at u. We again separate into two cases. Let y be the child of x that is an ancestor of j. In the first case, we let k be in the subtree rooted at y. Then, by our choice of i, we have

$$\mu_k(P^{t+1}) \sqsupseteq_k (\alpha, v) \sqsupseteq_k (\alpha, x) = (\alpha, \sigma_k(P^{t+1}_{-k}, \alpha)),$$

which, by Lemma 1, implies that statement 1 holds for k. In the second case, let k be in the subtree rooted at u but not in the subtree rooted at y. Again, $\sigma_k(P^{t+1}_{-k}, \gamma) = \sigma_k(P^t_{-k}, \gamma)$ for $\gamma \neq \alpha$. For α we have

$$(\alpha, \mathrm{lca}(k, \sigma_k(P^{t+1}_{-k}, \alpha))) = (\alpha, \mathrm{lca}(k, i)) \sqsupseteq_k (\alpha, x) \sqsupseteq_k \mu_k(P^t) = \mu_k(P^{t+1}),$$

establishing statement 1 for k using Lemma 1.

We finally establish statement 3. The fact $|W_{t+1}| \leq |W_t|$ is immediate from the definition of step t of the algorithm. When S is empty, $|W_{t+1}| < |W_t|$ since a fictional player is deleted. When S is nonempty, we have shown above that $\mu_i(P^{t+1}) \sqsupset_i \mu_i(P^t)$, thus completing the proof for statement 3.

Theorem 1. *Equilibria for hierarchical networks can be found in poly-time.*

Proof. It is immediate from the definition of the algorithm and Lemma 2 that at termination, the algorithm returns a valid equilibrium. We now show that our algorithm terminates in polynomial time. Consider the potential given by the sum of $|W_t|$ and the sum, over all i, of the position of $\mu_i(P^t)$ in the preorder $\sqsupseteq_i$. The term $|W_0|$ is at most nm, where n is $|V|$ (which is at least the number of internal nodes) and m is the number of objects. Furthermore, since $|\mathrm{O} \times I|$ is at most nm, the initial potential is at most $nm + n^2m$. By Lemma 2, the potential decreases by at least one in each step of the algorithm. Thus, the number of steps of the algorithm is at most $nm + n^2m$.

We now show that each step of the algorithm can be implemented in polynomial time. The initialization consists of adding the $O(nm)$ fictional players and computing the best response for each node i in V; the latter task involves, for each k in V, comparing at most m placements (one for each object). Each subsequent step of the algorithm involved the selection of a fictional player j, determination whether the set S is nonempty, and if so, computation of the node i, and then updating the placement. The only parts that need explanation are the computation of S and i; S is the set of all nodes k that are not in equilibrium when fictional player j is deleted. We compute S as follows: for each node k in V, if replacing the current object in their cache by α yields a more preferable placement then add k to S. Thus, S can be computed in polynomial time. The node i is simply a node in S such that $\mathrm{lca}(i, j)$ is lowest among all nodes in S, and can be computed in polynomial time.

4 A general axiomatic framework for CSR games

We now present a new axiomatic framework which generalizes the result of Section 3 to a broad class of utility functions.

Node preference relations. We assume that each node i in V has a total preorder $\geq_i$ among all the nodes in V[6]; $\geq_i$ further satisfies $i \geq_i j$ for all $i, j \in V$. We say that a node i *prefers* j over k if $j \geq_i k$, and call a node j *most i-preferred* in a set S of nodes if j is in S and $j \geq_i k$ for all k in S. We also use the notation $j =_i k$ whenever $j \geq_i k$ and $k \geq_i j$, and $j >_i k$ whenever it is not the case that $k \geq_i j$. Note that $>_i$ is a strict weak order[7], and for any i, j, and k, we have exactly one of these three relations holding: $j >_i k$, $k >_i j$, $k =_i j$. We also extend the notation $\sigma_i(P, \alpha)$ and $\sigma_i(P_{-i}, \alpha)$ denote a most i-preferred node holding α in P and P_{-i}, respectively, breaking ties arbitrarily.

Utility preference relations. In our game-theoretic model, each node attaches a utility to each global placement. We present a general definition that allows us to consider a large class of utility functions simultaneously. (The notation $\succ_i$ and $=_i$ over global placements are defined analogously.) We require that $\succeq_i$, for each $i \in V$, satisfies the following two basic conditions. [12] elaborates further on these conditions and their generality.

- **Monotonicity**: For any two global placements P and Q, if, for each object α and each node q with $\alpha \in Q_q$, there exists a node p with $\alpha \in P_p$ and $p \geq_i q$, then $P \succeq_i Q$.
- **Consistency**: Let (P_i, P_{-i}) and (Q_i, Q_{-i}) denote two global placements such that for each object $\alpha \in P_i \cup Q_i$, if p (resp., q) is a most i-preferred node in $V \setminus \{i\}$ holding α, i.e., $\alpha \in P_p$ (resp., $\alpha \in Q_q$), then $p =_i q$. If $(P_i, P_{-i}) \succ_i (Q_i, P_{-i})$, then $(P_i, Q_{-i}) \succeq_i (Q_i, Q_{-i})$.

[6] A total preorder is a binary relation that satisfies reflexivity, transitivity, and totality. Totality means that for any i, j, k, either $j \geq_i k$ or $k \geq_i j$.

[7] A strict weak order is a strict partial order $>$ (a transitive relation that is irreflexive) in which the relation "neither $a > b$ nor $b > a$" is transitive. Strict weak orders and total preorders are widely used in microeconomics.

Binary object preferences. One class of utility preference relations that we highlight is the ones based on binary object preferences. Suppose that each node i has a set S_i of objects in which it is equally interested, and it has no interest in the other objects. Let $\tau_i(P)$ denote the $|S_i|$-length sequence consisting of the $\sigma_i(P, \alpha)$, for $\alpha \in S_i$, in nonincreasing order according to the relation $\geq_i$. Then, the consistency condition can be further strengthened to the following.

– **Binary Consistency:** For any placements $P = (P_i, P_{-i})$ and $Q = (Q_i, Q_{-i})$ with $P_{-i} = Q_{-i}$, we have $P \succeq_i Q$ if and only if for $1 \leq k \leq |S_i|$, the kth component of $\tau_i(P)$ is at least as i-preferred as the kth component of $\tau_i(Q)$.

CSR **Games.** In the general framework, a CSR game is a tuple $(V, \mathrm{O}, \{\geq_i\}, \{\succeq_i\})$. A (pure) Nash equilibrium for an CSR game instance is a global placement P such that for each $i \in V$ there is no placement Q_i such that $(Q_i, P_{-i}) \succ_i (P_i, P_{-i})$. We argue in [12] that the unit cache capacity assumption of Section 2 continues to hold without loss of generality.

For our complexity results, we need to give the specification for a given game instance. The set V is specified, together with node cache capacities, and O is an enumerated list of object names. The node preference relation $\geq_i$ is specified succinctly by a set of at most $\binom{n}{2}$ bits, for each i. The utility preference relation $\succeq_i$, however, is over a potentially exponential number of placements (in terms of n, m, and cache sizes). For our complexity results, we assume that the utility preference relations are specified by an efficient algorithm – which we call the *utility preference oracle* – that takes as input a node i, and two global placements P and Q, and returns whether $P \succeq_i Q$. For the sum, max, and L_p-norm utilities, the utility preference oracle simply computes the relevant utility function. For binary object preferences, the binary consistency condition yields an oracle that is polynomial in number of nodes, objects, and cache sizes.

5 Existence of equilibria in the general framework

In this section, we show that equilibria exist for several CSR games under the axiomatic framework of Section 4. We first extend our result for sum utilities on hierarchical networks to the general framework. We next show that CSR games for undirected networks and binary object preferences are potential games. Finally, for the case of two objects, we give a polynomial-time construction of equilibria for CSR games for undirected networks. All proofs are deferred to [12].

Hierarchical networks. We show that the polynomial time algorithm of Section 3 extends to the axiomatic framework we have introduced. In the general framework, a hierarchical network can be represented as a tree T whose set of leaves is the node set V and the node preference relation $\geq_i$ given by: $j \geq_i k$ if $\mathrm{lca}(i, j)$ is a descendant of $\mathrm{lca}(i, k)$. Our algorithm of Section 3 and its analysis are completely determined by the structure of the hierarchical network and the pair-preference relations $\sqsupseteq_i$ defined for each node i; the latter were defined for the sum utility function. In order to extend our analysis to the general framework, it suffices to derive a new pair preference relation satisfying Lemma 1, which we now present for arbitrary utility preference relations satisfying the monotonicity and consistency properties.

Given any utility preference relation $\succeq_i$ that satisfies the monotonicity and consistency conditions, we define a strict weak order $\sqsupset_i$ on $\mathrm{O} \times A_i$, where A_i is the set of proper ancestors of i in T.

1. For each object α, node i, and proper ancestors v and w of i, we have $(\alpha, v) \sqsupset_i (\alpha, w)$ whenever v is a proper ancestor of w.
2. For objects α, β and nodes i, j, k with $\alpha \neq \beta$, $j, k \neq i$, let $\mathcal{P}$ be the set of global placements P such that j (resp., k) is a most i-preferred node in $V \setminus \{i\}$ holding α (resp., β) in P_{-i}. If *there exist* global placements $P = (\{\alpha\}, P_{-i})$ and $Q = (\{\beta\}, P_{-i})$ in $\mathcal{P}$ with $P \succ_i Q$, then $(\alpha, \mathrm{lca}(i, j)) \sqsupset_i (\beta, \mathrm{lca}(i, k))$.

In [12], we elaborate on the above definition, show that $\sqsupset_i$ is a well-defined strict weak order and also establish Lemma 1. The remainder of the analysis for hierarchical networks (Lemma 2 and Theorem 1) follows as before.

Undirected networks with binary object preferences. Let d be a symmetric cost function for an undirected network over the node set V. Recall that for binary object preferences, we are given, for each node i a set S_i of objects in which i is equally interested. Our proof of existence of equilibria is via a potential function argument. Given a placement P, let $\Phi_i(P) = d_{ij}$, where j is the most i-preferred node in $V - \{i\}$ holding the object in P_i. We introduce the potential function Φ: $\Phi(P) = (\Phi_0, \Phi_{i_1}(P), \Phi_{i_2}(P), \ldots, \Phi_{i_n}(P))$, where Φ_0 is the number of nodes i such that $P_i \subseteq S_i$, and $\Phi_{i_j}(P) \leq \Phi_{i_{j+1}}(P)$, $\forall j$, where $V = \{i_1, i_2, \ldots, i_n\}$. In [12], we prove that Φ is an increasing potential function: after any better response step, Φ increases in lexicographical order.

Undirected networks with two objects. We give a polynomial-time algorithm for computing an equilibrium in any undirected network with two objects. Our algorithm uses the fictional player technique introduced in Section 3. It starts by introducing fictional players serving both the objects in the network at zero cost from each node. In each subsequent step, we move the fictional players progressively "further" away, ensuring that each instant, we have an equilibrium. Finally, when the fictional players are at least preferred cost from all the nodes, they can be removed yielding an equilibrium for the original network.

6 Non-Existence of equilibria in CSR games and the associated decision problem

In this section, we show that the classes of games studied in Section 5 are essentially the only games where equilibria are guaranteed to exist, and study the complexity of the associated decision problem. All proofs are deferred to [12].

Theorem 2. *It is NP-hard to determine whether an* CSR *intance has an equilibrium even if one of these three restrictions hold: (a) number of objects is two; (b) object preferences are binary and number of objects is three; (c) network is undirected and number of objects is three.*

The proof is by a reduction from 3SAT [10]. Each reduction uses a gadget which has an equilibrium iff a specified node holds a certain object. Several copies of these gadgets are then put together to capture the given 3SAT formula.

Consider the problem 2BIN: does a given CSR instance with two objects and binary preferences possess an equilibrium? We prove that 2BIN is polynomial-time equivalent to EVEN-CYCLE [28]: does a given digraph contain an even cycle? Despite intensive efforts, EVEN-CYCLE was open until [24] provided a tour de force polynomial-time algorithm. Our result, thus, places 2BIN in P.

Theorem 3. EVEN-CYCLE *is polynomial-time equivalent to* 2BIN.

7 Concluding remarks

In this paper we have defined a capacitated replication game framework in networks, where the cache capacity of each node is bounded and all objects are of uniform size. We have shown that a pure Nash equilibrium can be computed for every hierarchical network, using a new notion of fictional players. In general, we have almost completely characterized the complexity of CSR games: For what classes of games do equilibria exist? Can we determine efficiently whether they exist? When they do exist, can we efficiently find them? One complexity question that is still open is the case of undirected networks with binary preferences. We conjecture that finding equilbria in such games (which we prove are potential games) is PLS-hard. In general, we would like to study the convergence of the best response process for the cases of games where equilibria exist.

In the full paper [12], we also consider a fractional version of CSR games, where each node is allowed to store fractions of objects. In our framework, which can be implemented using erasure codes (e.g., see Digital Fountain [4, 25]), a node can satisfy an object access request by retrieving any set of fractions of the object as long as these fractions sum to at least one. We have shown that finding an equilibrium of a fractional CSR game is complete for PPAD.

Finally, we note that our model assumes that the sets of nodes, objects, and preference relations are all static. We believe our results will be meaningful for environments where these sets change infrequently. Developing better models for addressing more dynamic scenarios is an important practical research direction.

References

1. E. Angel, E. Bampis, G. G. Pollatos, and V. Zissimopoulos. Optimal data placement on networks with constant number of clients. *CoRR*, abs/1004.4420, 2010.
2. K. Arrow. *Social Choice and Individual Values*. Yale University Press, 1951.
3. I. D. Baev, R. Rajaraman, and C. Swamy. Approximation algorithms for data placement problems. *SIAM J. Comput.*, 38(4):1411–1429, 2008.
4. J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege. A digital fountain approach to reliable distribution of bulk data. In *SIGCOMM '98*, pages 56–67, 1998.
5. X. Chen, X. Deng, and S.-H. Teng. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM (JACM)*, 56(3), 2009.

6. B.-G. Chun, K. Chaudhuri, H. Wee, M. Barreno, C. H. Papadimitriou, and J. Kubiatowicz. Selfish caching in distributed systems: a game-theoretic analysis. In *PODC*, pages 21–30, 2004.
7. C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a Nash equilibrium. *STOC ACM*, pages 71–78, 2006.
8. N. R. Devanur, N. Garg, R. Khandekar, V. Pandit, A. Saberi, and V. V. Vazirani. Price of anarchy, locality gap, and a network service provider game. In *WINE*, pages 1046–1055, 2005.
9. A. Fabrikant, A. Luthra, E. Maneva, C. H. Papadimitriou, and S. Shenker. On a network creation game. In *PODC*, pages 347–351, 2003.
10. M. Garey and D. Johnson. *Computers and intractability.* Freeman Press, 1979.
11. M. X. Goemans, L. Li, V. S. Mirrokni, and M. Thottan. Market sharing games applied to content distribution in ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(5):1020–1033, 2006.
12. R. Gopalakrishnan, D. Kanoulas, N. N. Karuturi, C. P. Rangan, R. Rajaraman, and R. Sundaram. Cache me if you can: Capacitated selfish replication in networks. *CoRR abs/1007.2694*, 2011.
13. D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, 1988.
14. D. Karger, E. Lehman, T. Leighton, M. Levine, D. Lewin, and R. P. Abstract. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web. In *STOC ACM*, pages 654–663, 1997.
15. M. Korupolu, C. G. Plaxton, and R. Rajaraman. Placement algorithms for hierarchical cooperative caching. *Journal of Algorithms*, 38:260–302, 2001.
16. M. R. Korupolu and M. Dahlin. Coordinated placement and replacement for large-scale distributed caches. *IEEE Trans. Knowl. Data Eng.*, 14(6):1317–1329, 2002.
17. E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *STACS*, pages 404–413, March 1999.
18. N. Laoutaris, G. Smaragdakis, K. Oikonomou, I. Stavrakakis, and A. Bestavros. Distributed placement of service facilities in large-scale networks. In *INFOCOM*, pages 2144–2152, 2007.
19. N. Laoutaris, O. Telelis, V. Zissimopoulos, and I. Stavrakakis. Distributed selfish replication. *IEEE Trans. Parallel Distrib. Syst.*, 17(12):1401–1413, 2006.
20. A. Leff, J. L. Wolf, and P. S. Yu. Replication algorithms in a remote caching architecture. *IEEE Trans. Parallel Distrib. Syst.*, 4(11):1185–1204, Nov 1993.
21. N. Nisan, T. Roughgarden, É. Tardos, and V. V. Vazirani. *Algorithmic Game Theory.* Cambridge University Press, 2007.
22. C. H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *JCSS*, 48(3):498–532, 1994.
23. G. G. Pollatos, O. Telelis, and V. Zissimopoulos. On the social cost of distributed selfish content replication. In *Networking*, pages 195–206, 2008.
24. N. Robertson, P. D. Seymour, and R. Thomas. Permanents, pfaffian orientations, and even directed circuits. *Annals of Mathematics*, pages 929–975, 1999.
25. A. Shokrollahi. Raptor codes. In *IEEE Trans Inf Theory*, pages 2551–2567, 2006.
26. R. Tewari, M. Dahlin, H. M. Vin, and J. S. Kay. Design considerations for distributed caching on the internet. In *ICDCS*, pages 273–284, 1999.
27. O. Wolfson, S. Jajodia, and Y. Huang. An adaptive data replication algorithm. *ACM Transactions on Database Systems*, 22:255–314, 1997.
28. D. H. Younger. Graphs with interlinked directed circuits. In *Proceedings of Midwestern Symposium on Circuit Theory*, volume 2, pages XVI2.1–XVI2.7, 1973.