Cooperative-Game Framework for Resource Sharing in SDC

Faheem Zafari[1], Kin K. Leung[2], Don Towsley[3], Prithwish Basu[4], Ananthram Swami[5], and Jian Li[6]
[1,2]Imperial College London, [3,6]University of Massachusetts Amherst, [4]BBN Technologies,
[5]U.S. Army Research Laboratory,
[1,2]{faheem16, kin.leung}@imperial.ac.uk, [3,6]{towsley,jianli}@cs.umass.edu,
[4]prithwish.basu@raytheon.com, [5]ananthram.swami.civ@mail.mil

***Abstract*—Providing resources to users (applications) is fundamental to SDC for coalition operations. Allocating resources to users optimally is a challenging problem. As some domains (service providers) may not have adequate resources to satisfy all local users, providing resources to meet all user demands across all domains of the SDC is particularly challenging. Resource sharing among domains can improve resource availability and utilization as surplus of resources in some domains can be "rented" by other domains. However, different domains (and coalition partners) can have different objectives (utilities). Therefore, there is a need for a framework that can share and allocate resources in an efficient way, while considering the distinct objectives among domains. In this paper, we formulate the issue as a multi-objective optimization problem and propose a framework based on Cooperative Game Theory (CGT) for resource sharing and allocation among domains. We show that the resource-sharing problem can be modeled as an N-player canonical cooperative game with non-transferable utility (NTU) and prove that the game is convex. For two resource-sharing priority strategies for the framework, we propose efficient algorithms that provide resource allocations from the core, hence guaranteeing Pareto optimality. By simulation, we evaluate and show that the proposed framework can improve user satisfaction and domain utilities.**

I. Introduction

An SDC slice consists of multiple domains (service providers) that have different types of resources. Applications (users) can use these resources for a price, while domains are expected to provide resources according to a *Service Level Agreement* (SLA). Providing resources to applications has been an important area of research [1]–[4] as resources may not satisfy demand. Even when resources are available, allocating them to applications with the goal of maximizing the overall domain *utility* is a difficult problem. Furthermore, the advent of resource intensive paradigms such as deep learning, and data analytics exacerbate the problem due to the scalability of traditional resource allocation techniques.

This work was supported by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copy-right notation hereon. Faheem Zafari also acknowledges the financial support by EPSRC Centre for Doctoral Training in High Performance Embedded and Distributed Systems (HiPEDS, Grant Reference EP/L016796/1), and Department of Electrical and Electronics Engineering, Imperial College London.

In this paper, we consider an SDC slice consisting of different domains (service providers) belonging to different coalition partners. Each domain has a specific amount of resources and particular applications associated with the domain (referred to as the native applications) can request for resources. We consider two different strategies for resource allocation: 1) The domains do not differentiate between their native applications and applications of other domains, and 2) Domains first attempt to allocate resources to their native applications before considering applications from other domains. If a domain can satisfy its own applications and has resource surplus, it can share them with other domains that need resources. We present a resource sharing and allocation framework based on *cooperative game theory* (CGT), where domains share their resources and form a coalition to satisfy requests of different applications. Our CGT-based framework takes into account the fact that different domains may have different objectives, which is why a traditional single objective optimization framework cannot be applied.

A. Summary of the contributions

The contributions of this paper are:

1) We present a novel multi-objective resource sharing and allocation problem formulation that considers both service provider utilities and application request satisfaction in SDC. We propose a cooperative-game-theory based framework to solve the formulated problem. We model each domain as a player in our game and show that the resource sharing/allocation problem can be modeled as a *canonical* game with non-transferable utility (NTU). We prove that the game is super-additive and convex, hence the core is non-empty. Therefore, the grand coalition of all domains is stable.
2) We propose efficient algorithms for the aforementioned priority strategies that provide allocations from the core. Since the allocations are from the core, the obtained solutions are Pareto-optimal.
3) We evaluate the performance of our proposed framework through extensive simulations. We show that the resource sharing/allocation mechanism improves the utilities of game players. Furthermore, our framework also improves user satisfaction.

The paper is structured as follows: Section II provides a primer on CGT, and the core. Section III presents our system model. It also presents the resource sharing/allocation optimization problem. Section IV discusses the game theoretic solution. Section V presents our experimental results. Section VI presents a review of relevant literature, while Section VII concludes the paper.

II. Preliminaries

This section discusses briefly cooperative game theory and the core. Formally,
Formally,

Definition 1. *Coalition Games [5]: Any coalition game with* non-transferable utility *(discussed below) can be represented by the pair $(\mathcal{N}, \mathcal{V})$ where $\mathcal{N}$ is the set of players that play the game, while $\mathcal{V}$ is a set of payoff vectors such that [6]:*

1) *$\mathcal{V}(S)$ should be a closed and convex subset of $\mathbb{R}^S$.*
2) *$\mathcal{V}(S)$ should be comprehensive, i.e., if we are given payoffs $\mathbf{x} \in \mathcal{V}(S)$ and $\mathbf{y} \in \mathbb{R}^S$ where $\mathbf{y} \leq \mathbf{x}$, then $\mathbf{y} \in \mathcal{V}(S)$. In other words, if the members of coalition S can achieve a payoff allocation $\mathbf{x}$, then the players can change their strategies to achieve an allocation $\mathbf{y}$ where $\mathbf{y} \leq \mathbf{x}$.*
3) *The set $\{\mathbf{x}|\mathbf{x} \in \mathcal{V}(S) \text{ and } x_n \geq z_n, \forall n \in S\}$, with $z_n = \max\{y_n|\mathbf{y} \in \mathcal{V}(\{n\})\} \leq \infty \; \forall n \in \mathcal{N}$ should be a bounded subset of $\mathbb{R}^S$.*

Definition 2. *Value of a coalition: The total utility of any coalition $S \subseteq \mathcal{N}$ is known as the value of a coalition $v(S)$.*

Definition 3. *Non-Transferable Utility (NTU) [5]: If the total utility of any coalition $v(S)$ cannot be assigned a single real number or if there is a rigid restriction on utility distribution among players, then the game has a* non-transferable *utility.*

In other words, the payoff each player receives and the value of coalition depend on the game strategy. The utility cannot be transfered freely among the players.

Definition 4. *Characteristic function: The characteristic function for any coalition game with NTU is a function that assigns a set of payoff vectors, $\mathcal{V}(S) \subseteq \mathbb{R}^S$, to game players where each element of the payoff vector x_n represents a payoff that player $n \in S$ obtains within the coalition S given a strategy selected by the player n in the coalition S.*

Definition 5. *Characteristic Form Coalition Games [5]: A coalition game is said to be of* characteristic *form, if the value of coalition $S \subseteq \mathcal{N}$ depends only on the members of the coalition.*

Definition 6. *Superadditivity of NTU games: A canonical game with NTU is said to be superadditive if the following property is satisfied.*

$$\mathcal{V}(S_1 \cup S_2) \supset \{x \in \mathbb{R}^{S_1 \cup S_2} | (x_n)_{n \in S_1} \in \mathcal{V}(S_1), (x_m)_{m \in S_2} \in \mathcal{V}(S_2)\} \; \forall S_1 \subset \mathcal{N}, S_2 \subset \mathcal{N}, S_1 \cap S_2 = \emptyset. \quad (1)$$

where x is a payoff allocation for the coalition $S_1 \cup S_2$.

That is, if any two disjoint coalitions S_1 and S_2 form a large coalition $S_1 \cup S_2$, then the coalition $S_1 \cup S_2$ can always give its members the payoff that they would have received in the disjoint coalition S_1 and S_2. It is worth mentioning that $\mathcal{V}$ is the set of payoff vectors while v is the sum of payoffs for all coalition.

Definition 7. *Canonical Game: A coalition game is canonical if it is superadditive and in characteristic form.*

Definition 8. *Group Rational: A payoff vector $\boldsymbol{x} \in \mathbb{R}^{\mathcal{N}}$ for distributing the coalition value $v(\mathcal{N})$ to all players is group-rational if $\sum_{n \in \mathcal{N}} x_n = v(\mathcal{N})$.*

Definition 9. *Individual Rational: A payoff vector $\boldsymbol{x} \in \mathbb{R}^{\mathcal{N}}$ is individually rational if every player can obtain a payoff no less than not in a coalition, i.e., $x_n \geq v(\{n\}), \forall n \in \mathcal{N}$.*

Definition 10. *Imputation: A payoff vector that is both individual and group rational is an imputation.*

Definition 11. *Grand Coalition: The coalition formed by all game players $\mathcal{N}$ is the grand coalition.*

Based on the above definitions, we can define the core of an NTU canonical coalition game as:

Definition 12. *Core [5]: For any NTU canonical game $(\mathcal{N}, \mathcal{V})$, the core C_{NTU} is the set of imputations in which no coalition $S \subset \mathcal{N}$ has any incentive to reject the proposed payoff allocation and deviate from the grand coalition to form a coalition S instead. This can be mathematically expressed as*

$$\mathcal{C}_{NTU} = \{\mathbf{x} \in \mathcal{V}(\mathcal{N}) | \forall S \subset \mathcal{N}, \nexists \mathbf{y} \in \mathcal{V}(S) \; such \; that \; y_n > x_n, \forall n \in S\}. \quad (2)$$

However, the core is not always guaranteed to be non-empty. Even if the core exists, it either may be a single point or infinite number of points as it is a convex set, where very large means infinite number of points. Therefore, finding a payoff in the core is not easy.

III. System Model

Let $\mathcal{N} = \{1, 2, \cdots, N\}$ be the set of all service providers (SP)/domain. We assume that each domain has a set of $\mathcal{K} = \{1, 2, \cdots, K\}$ different types of resources such as communication, computation and storage resources. Domain n has resources $C_n = \{C_{n,1}, \cdots, C_{n,K}\}$, where $C_{n,k}$ is the amount of type k resources available at service provider n. Each domain n has a set of native (local) applications $\mathcal{M}_n = \{1, 2, \cdots, M_n\}$. The set of all applications that ask for resources from all domains is given by $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2 \cdots \cup \mathcal{M}_N$, where $\mathcal{M}_i \cap \mathcal{M}_j = \emptyset$, $\forall i \neq j$, i.e., each application originally asks only one domain for resources. $r_{n,k}^j$ is the amount of resource k that application $j \in \mathcal{M}_n$ requests from domain n. When the domain is working alone[1], its objective is to maximize its utility by allocating resources to its native applications and improving application satisfaction. A domain n earns a non-negative *monotonically non-decreasing* utility $u_n^j(x_{n,k}^j)$ by making an allocation decision $x_{n,k}^j$, i.e.,

[1]Not borrowing from or renting out its resources to any other domain.

allocating x amount of resource k to application $j \in \mathcal{M}_n$, where the vector $\mathbf{x}_n^j = [x_{n,1}^j, x_{n,2}^j, \cdots, x_{n,K}^j]^T$. Below we present the optimization formulation for a single domain.

A. Single Domain Problem Formulation

We first present the resource allocation problem for a single domain (no resource sharing with other domains). For each domain $n \in \mathcal{N}$, the allocation decisions consist of the vectors $\mathbf{x}_n^1, \cdots, \mathbf{x}_n^{M_n}$. The optimization problem is:

$$\max_{\mathbf{x}_n^1, \cdots, \mathbf{x}_n^{M_n}} \quad \sum_{j \in \mathcal{M}_n} f_n^j(\mathbf{x}_n^j), \tag{3a}$$

$$\text{s.t.} \quad \sum_j x_{n,k}^j \leq C_{n,k}, \quad \forall k \in \mathcal{K} \tag{3b}$$

$$x_{n,k}^j \leq r_{n,k}^j, \quad \forall\, j \in \mathcal{M}_n, k \in \mathcal{K}, \tag{3c}$$

$$x_{n,k}^j \geq 0, \quad \forall\, j \in \mathcal{M}_n, k \in \mathcal{K}. \tag{3d}$$

The goal of this single objective optimization problem for every domain, as mentioned earlier, is to maximize its utility by allocating available resources only to its own applications and increasing application satisfaction. The objective function $f_n^j(\mathbf{x}_n^j)$ is a monotonic non-decreasing function. In this paper, we assume that:

$$f_n^j(\mathbf{x}_n^j) = u_n^j(\mathbf{x}_n^j) + \sum_{k \in \mathcal{K}} \frac{x_{n,k}^j}{r_{n,k}^j},$$

where the term $\sum_{j \in \mathcal{M}_n} \sum_{k \in \mathcal{K}} \frac{x_{n,k}^j}{r_{n,k}^j}$ captures the satisfaction of resource requests by applications and $u_n^j(\mathbf{x}_n^j)$ is the utility paid by applications to domains for allocating resources. The first constraint in (3b) indicates that the allocated resources cannot exceed capacity. The second constraint in (3c) indicates that the allocated resources should not exceed required resources. The last constraint, (3d) indicates that allocation cannot be negative. However, it is possible that a domain n may earn a larger utility by providing its resources to applications of other domains or it may not have sufficient resources to satisfy requests of all its native applications. On the other hand, there may be other geographically co-located domain $m \in \mathcal{N} \backslash n$ that may have resource surpluses, which can be "rented" by domain n. In this work, we focus on the strategy where each domain has to allocate resources to its native applications and then share the remaining (if any) resources with other domains who are not able to satisfy their native applications. Below, we discuss resource sharing among the aforementioned domains.

B. Multiple Domains Problem Formulation

Let's assume that a domain n does not have sufficient resources to satisfy all its native applications. A domain m may have resource surplus, after allocating its resources to its native applications, that it can share with n. Therefore, allowing resource sharing among such domains, while considering their objectives can improve resource utilization and improve application satisfaction. By sharing its resources, player m earns a non-decreasing and non-negative utility $H_m^l(x_{m,k}^l)$, which in this paper is assumed to be as follows [2].

$$H_m^l(x_{m,k}^l) = u_m^j(x_{m,k}^l) - C_m^l(x_{m,k}^l), \forall l \in \mathcal{M} \backslash \mathcal{M}_m, \tag{4}$$

whereas $C_m^j(x_{m,k}^j)$ captures the communication cost of serving application $j \in \mathcal{M}_n$ at domain m rather than its native domain n. Also, $F_n^j(\mathbf{x}_n^j)$ represents the utility a domain n earns by allocating resources its native application when sharing resources with other domains. We restrict $F_n^j(\mathbf{x}_n^j)$ to be a non-negative and non-decreasing function and assume:

$$F_n^j(\mathbf{x}_n^j) = u_n^j(\mathbf{x}_n^j) + \sum_{k \in \mathcal{K}} \frac{\sum_{m \in \mathcal{N}} x_{m,k}^j}{r_{n,k}^j}, \tag{5}$$

We also assume that $u_n^j(x_{n,k}^j) = u_m^j(x_{m,k}^j)$ if and only if $x_{n,k}^j = x_{m,k}^j$. Furthermore,

$$\mathbf{x}^j = [\sum_{n \in \mathcal{N}} x_{n,1}^j, \sum_{n \in \mathcal{N}} x_{n,2}^j, \cdots, \sum_{n \in \mathcal{N}} x_{n,K}^j]^T,$$

i.e., the total resource allocated to any application $j \in \mathcal{M}$ is the sum of resource allocated to it across all domains. The resource sharing and allocation framework, based on resource requests and capacities of domains, has to make an allocation decision that optimizes the utilities of all the domains $n \in \mathcal{N}$ and satisfy user requests as well. The allocation decision is given by $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_N\}$, where $\mathbf{X}_n, \ \forall n \in \mathcal{N}$ is given by:

$$\mathbf{X}_n = \begin{bmatrix} \mathbf{x}_n^1 \\ \vdots \\ \mathbf{x}_n^{|\mathcal{M}|} \end{bmatrix} = \begin{bmatrix} x_{n,1}^1 & \cdots & \cdots & x_{n,K}^1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{n,1}^{|\mathcal{M}|} & \cdots & \cdots & x_{n,K}^{|\mathcal{M}|} \end{bmatrix} \tag{6}$$

When domains are sharing resources, each domain aims to maximize sum of utilities obtained by a) allocating resources to its native applications; b) allocating resources (if available) to applications of other domains; and c) improving its applications' satisfaction by using its own resources or borrowing from other domains (if needed). In the resource sharing case, each domain solves the optimization problem given in (7).

$$\max_{\mathbf{X}_n} \quad \sum_{j \in \mathcal{M}_n} F_n^j(\mathbf{x}_n^j) + \sum_{l \in \{\mathcal{M} \backslash \mathcal{M}_n\}} H_n^l(\mathbf{x}_n^l), \quad \forall n \in \mathcal{N}, \tag{7a}$$

$$\text{s.t.} \quad \sum_j x_{n,k}^j \leq C_{n,k}, \quad \forall\, j \in \mathcal{M}, k \in \mathcal{K}, n \in \mathcal{N}, \tag{7b}$$

$$\sum_{m \in \mathcal{N}} x_{m,k}^j \leq r_{n,k}^j, \quad \forall\, j \in \mathcal{M}, k \in \mathcal{K}, n \in \mathcal{N}, \tag{7c}$$

$$x_{n,k}^j \geq 0, \quad \forall\, j \in \mathcal{M}, k \in \mathcal{K}, n \in \mathcal{N}, \tag{7d}$$

$$u_n^l(x_{n,k}^l) \geq C_n^l(x_{n,k}^l), \quad \forall l \in \mathcal{M} \backslash \mathcal{M}_n, k \in \mathcal{K}, n \in \mathcal{N}. \tag{7e}$$

[2] Our framework can work with any non-negative, non-decreasing utility.

Plugging (4) and (5) into (7a), the objective function can be rewritten as:

$$\sum_{j\in\mathcal{M}_n}\big(u_n^j(\mathbf{x}_n^j)+\sum_{k\in\mathcal{K}}\frac{\sum_{m\in\mathcal{N}}x_{m,k}^j}{r_{n,k}^j}\big)+ \\ \big(\sum_{l\in\{\mathcal{M}\setminus\mathcal{M}_n\}}u_n^l(\mathbf{x}_n^l)-C_n^l(\mathbf{x}_n^l)\big),\quad \forall n\in\mathcal{N}, \tag{8}$$

As the utility $u_n^j,\ \forall n\in\mathcal{N}$ can be different for each application j, (7) is a multi-objective optimization problem. Constraint (7b) indicates that total resources allocated cannot be more than the capacity of domains. (7c) means that the total amount of resources allocated to any application using the resource sharing framework cannot be more than the amount of requested resources, whereas (7d) indicates that the allocation decision cannot be negative. The final constraint (7e) indicates that the utility earned by sharing resources is more than the communication cost.

In this paper, we assume that the utility of each domain and the communication cost (C_m^j) is a non-negative and non-decreasing monotone function. We also assume that a subset $\mathcal{G}_1\subset\mathcal{N}$ of domains has resource deficit, while a subset $\mathcal{G}_2\subset\mathcal{N}$ has a resource surplus. In the next section, we present a game theoretic solution for resource sharing among different domains.

IV. Game Theoretic Solution

In this section, we first present general properties of our game and show that the resource sharing problem can be modeled as a canonical cooperative game with non-transferable utility. We then show that the core is non-empty, by proving that our canonical game is *convex*. To address the problem of obtaining an allocation from the core for both strategies, we propose algorithms in Section IV-B that provide allocations from the core for each strategy.

A. General Game Properties

We model each domain as a player in our game to obtain the optimal resource sharing and allocation decision. Let $\mathcal{N}$ be the set of players that can play the resource sharing and allocation game. The *value of coalition* for the game players $S\subseteq\mathcal{N}$ is given by (10), where $\mathcal{F}_S$ is the feasible set for resource sharing and allocation given by (7b)-(7d) and $\mathcal{M}_S$ is the set of applications in S.

$$v(S)=\sum_{\substack{n\in S\\ \mathcal{X}\in\mathcal{F}_S}}\Bigg(\sum_{j\in\mathcal{M}_n}\big(u_n^j(\mathbf{x}_n^j)+\sum_{k\in\mathcal{K}}\frac{\sum_{m\in S}x_{m,k}^j}{r_{n,k}^j}\big)+ \\ \big(\sum_{l\in\{\mathcal{M}\setminus\mathcal{M}_n\}}u_n^l(\mathbf{x}_n^l)-C_n^l(\mathbf{x}_n^l)\big)\Bigg) \tag{10}$$

Our strategy to achieve the aforementioned value of coalition is that each player first allocates resources to its own applications and then shares the remaining resources (if any) with applications of other players.

Remark 1. *Resource allocation and sharing problem (with multiple objectives) for the aforementioned system model can be modeled as a canonical cooperative game with NTU.*

This is because our characteristic function of resource sharing and allocation problem satisfies the following two conditions.

- **Characteristic form of payoff:** As the utility function in a resource sharing and allocation problem only relies on the service providers that are part of the coalition, the game or payoff function is of characteristic form.
- **Superadditivity:** Let $S_1,S_2\subseteq\mathcal{N}$ and $S_1\cap S_2=\emptyset$. Hence, $S_1,\ S_2\subset(S_1\cup S_2)$. Superadditivity follows from definition of monotone utilities.

Definition 13 (Convex Games)**.** *A coalition game is said to be convex if and only if for every player $n\in\mathcal{N}$, the marginal contribution of the player is non-decreasing W.R.T set inclusion. Mathematically, for $S_1\subseteq S_2\subseteq\mathcal{N}\setminus\{n\}$*

$$v(S_1\cup\{n\})-v(S_1)\le v(S_2\cup\{n\})-v(S_2). \tag{11}$$

Theorem 1. *Our canonical game is convex.*

Proof. Let us consider two coalitions S_1 and S_2, where $S_1\subseteq S_2\subseteq\mathcal{N}\setminus\{t\}$, $t\in\mathcal{N}$, and $\mathcal{F}_{S1}$ and $\mathcal{F}_{S2}$ are the feasible sets for S_1 and S_2 respectively that it achieves by allocating resources to its own applications. We calculate $v(S_2\cup\{t\})-v(S_1\cup\{t\})$ in Equation (9).

The last inequality in (9) follows from the fact that $\big(\sum_{l\in\{\mathcal{M}_{S_2}\setminus\mathcal{M}_{S_1}\}}u_t^l(\mathbf{x}_t^l)-C_t^l(\mathbf{x}_t^l)\big)+\sum_{j\in\mathcal{M}_t}\sum_{k\in\mathcal{K}}\frac{\sum_{m\in(\{S_1\setminus S_2\})}x_{m,k}^j}{r_{t,k}^j}\ \ge\ 0$ as no rational player will share resources if the communication cost $C_t^l(\mathbf{x}_t^l)$ is more than the utility $u_t^l(\mathbf{x}_t^l)$.

□

Remark 2. *The core of any convex game $(\mathcal{N},\mathcal{V})$ is non-empty and large [7].*

Lemma 1. *Our canonical cooperative game $(\mathcal{N},\mathcal{V})$ with NTU can be used to obtain the Pareto-optimal solutions for the multi-objective resource sharing and allocation among different service providers (domains).*

The convexity of our game proves that the core is non-empty. However, obtaining an allocation from the core is challenging that we address in the following subsection.

B. Proposed Algorithms

1) Strategy 1: In algorithm 1, we first solve the single objective optimization problem (3) for all players $n\in\mathcal{N}$. We then solve the problem in (12) that provides the allocation from the core. $v(\{n\})$ in (12b) is the payoff a player n receives when working alone.

Theorem 2. *The allocation decision obtained using Algorithm 1 lies in the core.*

Proof. To prove the theorem, we need to show that the allocation decision obtained using Algorithm 1: a) is individually

$$
\begin{aligned}
&v(S_2 \cup \{t\}) - v(S_1 \cup \{t\}) = \sum_{\substack{n \in S_2, \\ \mathcal{X} \in \mathcal{F}_{S2}}} \Big(\sum_{j \in \mathcal{M}_n} \big(u_n^j(\mathbf{x}_n^j) + \sum_{k \in \mathcal{K}} \frac{\sum_{m \in (S_2 \cup \{t\})} x_{m,k}^j}{r_{n,k}^j}\big) + \big(\sum_{\substack{l \in \{\mathcal{M}_{S_2} \cup \mathcal{M}_t\}, \\ l \notin \mathcal{M}_n}} u_n^l(\mathbf{x}_n^l) - C_n^l(\mathbf{x}_n^l)\big)\Big) + \\
&\sum_{j \in \mathcal{M}_t} \big(u_t^j(\mathbf{x}_t^j) + \sum_{k \in \mathcal{K}} \frac{\sum_{m \in (S_2 \cup \{t\})} x_{m,k}^j}{r_{t,k}^j}\big) + \big(\sum_{l \in \mathcal{M}_{S_2}} u_t^l(\mathbf{x}_t^l) - C_t^l(\mathbf{x}_t^l)\big) - \Big(\sum_{\substack{n \in S_1, \\ \mathcal{X} \in \mathcal{F}_{S1}}} \Big(\sum_{j \in \mathcal{M}_n} \big(u_n^j(\mathbf{x}_n^j) + \sum_{k \in \mathcal{K}} \frac{\sum_{m \in (S_1 \cup \{t\})} x_{m,k}^j}{r_{n,k}^j}\big) \\
&+ \big(\sum_{\substack{l \in \{\mathcal{M}_{S_1} \cup \mathcal{M}_t\}, \\ l \notin \mathcal{M}_n}} u_n^l(\mathbf{x}_n^l) - C_n^l(\mathbf{x}_n^l)\big)\Big) + \sum_{j \in \mathcal{M}_t} \big(u_t^j(\mathbf{x}_t^j) + \sum_{k \in \mathcal{K}} \frac{\sum_{m \in (S_1 \cup \{t\})} x_{m,k}^j}{r_{t,k}^j}\big) + \big(\sum_{l \in \mathcal{M}_{S_1}} u_t^l(\mathbf{x}_t^l) - C_t^l(\mathbf{x}_t^l)\big)\Big) \\
&v(S_2 \cup \{t\}) - v(S_1 \cup \{t\}) = \sum_{\substack{n \in S_2 \backslash S_1, \\ \mathcal{X} \in \mathcal{F}_{S2} \backslash \mathcal{F}_{S1}}} \Big(\sum_{j \in \mathcal{M}_n} u_n^j(\mathbf{x}_n^j) + \big(\sum_{\substack{l \in \{(\mathcal{M}_{S_2} \cup \mathcal{M}_t) \backslash (\mathcal{M}_{S_1} \cup \mathcal{M}_t)\}, \\ l \notin \mathcal{M}_n}} u_n^l(\mathbf{x}_n^l) - C_n^l(\mathbf{x}_n^l)\big) + \\
&\sum_{j \in \mathcal{M}_n} \sum_{k \in \mathcal{K}} \frac{\sum_{m \in \{(S_2) \backslash (S_1)\}} x_{m,k}^j}{r_{n,k}^j}\Big) + \big(\sum_{l \in \{\mathcal{M}_{S_2} \backslash \mathcal{M}_{S_1}\}} u_t^l(\mathbf{x}_t^l) - C_t^n(\mathbf{x}_t^l)\big) + \sum_{j \in \mathcal{M}_t} \sum_{k \in \mathcal{K}} \frac{\sum_{m \in (\{S_1 \backslash S_2\})} x_{m,k}^j}{r_{t,k}^j} \\
&v(S_2 \cup \{t\}) - v(S_1 \cup \{t\}) = \quad v(S_2) - v(S_1) + \big(\sum_{l \in \{\mathcal{M}_{S_2} \backslash \mathcal{M}_{S_1}\}} u_t^l(\mathbf{x}_t^l) - C_t^l(\mathbf{x}_t^l)\big) + \sum_{j \in \mathcal{M}_t} \sum_{k \in \mathcal{K}} \frac{\sum_{m \in (\{S_1 \backslash S_2\})} x_{m,k}^j}{r_{t,k}^j} \\
&v(S_2 \cup \{t\}) - v(S_1 \cup \{t\}) \geq \quad v(S_2) - v(S_1) \qquad (9)
\end{aligned}
$$

rational; b) is group rational; and c) no players have the incentive to leave the grand coalition and form another sub-coalition $S \subset \mathcal{N}$.

Individual Rationality: For each player $n \in \mathcal{N}$, the solution obtained using (12) is individual rational due to the constraint in (12b). Hence the solution obtained as a result of Algorithm 1 is individually rational.

Group Rationality: The value of the grand coalition $v\{\mathcal{N}\}$ as per Equation (12a) is the sum of utilities of all players that they achieve from the pay-off vector $\mathcal{V}(\mathcal{N}) \subseteq \mathbb{R}^{|\mathcal{N}|}$. Hence the allocation obtained from Algorithm 1 is group rational.

Furthermore, due to super-additivity of the game and monotonic non-decreasing nature of the utilities, no subgroup of players have an incentive to form a smaller coalition. Hence Algorithm 1 provides a solution from the core. □

$$
\begin{aligned}
\max_{\mathbf{X}} \quad & \sum_{n \in \mathcal{N}} \Big(\sum_{j \in \mathcal{M}_n} \big(u_n^j(\mathbf{x}_n^j) + \sum_{k \in \mathcal{K}} \frac{\sum_{m \in \mathcal{N}} x_{m,k}^j}{r_{n,k}^j}\big) + \\
& \big(\sum_{l \in \{\mathcal{M} \backslash \mathcal{M}_n\}} u_n^l(\mathbf{x}_n^l) - C_n^l(\mathbf{x}_n^l)\big), \Big), \qquad (12a) \\
\text{s.t.} \quad & \texttt{constraints in } (7b)\text{–}(7d), \\
& \Big(\sum_{j \in \mathcal{M}_n} \big(u_n^j(\mathbf{x}_n^j) + \sum_{k \in \mathcal{K}} \frac{\sum_{m \in \mathcal{N}} x_{m,k}^j}{r_{n,k}^j}\big) + \\
& \big(\sum_{l \in \{\mathcal{M} \backslash \mathcal{M}_n\}} u_n^l(\mathbf{x}_n^l) - C_n^l(\mathbf{x}_n^l)\big), \Big) \geq v(\{n\}), \forall n \in \mathcal{N}, \qquad (12b) \\
& u_n^l(x_{n,k}^l) \geq C_n^l(x_{n,k}^l), \quad \forall l \in \mathcal{M} \backslash \mathcal{M}_n, k \in \mathcal{K}, n \in \mathcal{N}. \qquad (12c)
\end{aligned}
$$

Algorithm 1 Pareto optimal allocation for strategy 1

Input: R, C, and vector of utility functions of all players $\mathbf{u}$
Output: $\mathbf{X}$, $\mathbf{u}(\mathbf{X})$
Step 1**:** $\mathbf{u}(\mathbf{X}) \leftarrow 0$, $\mathbf{X} \leftarrow 0$
Step 2**:**
for $n \in \mathcal{N}$ **do**
  $v(\{n\})$ ←`Optimal objective function value in` (3)
end for
Step 3**:** $\mathbf{X}$ ←`Optimal allocation decision from` (12)
  $\mathbf{u}(\mathbf{X})$ ←`Payoff vector from` (12)

2) Strategy 2: We propose a computationally efficient algorithm in Algorithm 2 that requires solving $N + |\mathcal{G}_2|$ optimization problems. The input into the algorithm are available resources/capacities, user requirements and the utilities of all game players. The output of the algorithm is allocation decision. In Step 1, the utilities of the players, the allocation decision, and vectors $\boldsymbol{v}$, $\boldsymbol{O}^1$ and $\boldsymbol{O}^2$ are initialized. In Step 2, every player's utility is calculated without any resource sharing, i.e., the single objective optimization problem is solved by every player and the values are stored in the vector $\boldsymbol{O}^1$. Then C, R are updated and stored in C' and R', respectively. We then move to Step 4, where the players are divided on the basis of updated capacities and requests into two groups $\mathcal{G}_1$ (resource deficit) and $\mathcal{G}_2$ (resource surplus) respectively. In Step 5, we obtain resource allocation decision and calculate the utility players in $\mathcal{G}_2$ earn by sharing their resources with players in $\mathcal{G}_1$. In step 6, the utility reflecting an increase in user satisfaction for native applications of players in $\mathcal{G}_1$ is obtained using the allocation decision from Step 5.

Algorithm 2 Pareto optimal allocation for strategy 2

Input: R, C, and vector of players' utility function $\mathbf{u}$
Output: $\mathbf{X}$
Step 1: $\mathbf{u}(\mathbf{X}) \leftarrow 0$, $\mathbf{X} \leftarrow 0$, $\boldsymbol{v} \leftarrow 0$, $\boldsymbol{O}^1 \leftarrow 0$, $\boldsymbol{O}^2 \leftarrow 0$
Step 2:
for $n \in \mathcal{N}$ **do**
  $v(\{n\}) \leftarrow$ `Objective function value of` (3)
  $\mathbf{x}_n^1, \cdots, \mathbf{x}_n^{M_n} \leftarrow$ `Allocation decision of` (3)
end for
Step 3: Update C and R based on Step 2
  $C' \leftarrow C_{updated}$, $R' \leftarrow R_{updated}$
Step 4: Divide the players into two subsets $\mathcal{G}_1$ and $\mathcal{G}_2$ representing players with resource deficit and resource surplus
Step 5:
for $n \in \mathcal{G}_2$ **do**
  $O_n^1 \leftarrow$ `Optimal objective function value of` (13)
  $\mathcal{X}_n \leftarrow$ `Optimal allocation decision of` (13)
  Update C' and R'
end for
Step 6:
for $m \in \mathcal{G}_1$ **do**
  $O_m^2 \leftarrow$ `Utility earned by satisfying users` $j \in \mathcal{M}_m$ `due to allocation decision in Step 5`
end for

Theorem 3. *The solution obtained from Algorithm 2 lies in the* core.

Proof. To prove the theorem, we need to show that the utilities obtained in Step 5 of Algorithm 2; a) are individually rational; b) are group rational; and c) no group of players has the incentive to leave the grand coalition to form another sub-coalition $S \subset \mathcal{N}$.

Individual Rationality: For each player $n \in \mathcal{N}$, the solution $v(\{n\})$ obtained by solving the optimization problem in (3) is the utility a player will obtain by working alone. The value O_n^1 for $n \in \mathcal{G}_2$ in Step 5 is either zero or positive but cannot be negative due to the nature of the utility used. Furthermore for players in $\mathcal{G}_1$, their utility may increase due to increase in user satisfaction if its users are provided resources by players in $\mathcal{G}_2$ given by O_n^2. Hence the solution obtained as a result of Algorithm 2 is individually rational.

Group Rationality: The value of the grand coalition $v\{\mathcal{N}\}$ as per Equation (10) is the sum of utilities. Steps 2, 5 and 6 of Algorithm 2 obtain the sum of utilities as well. Hence the solution obtained as a result of Algorithm 2 is group rational. Furthermore, due to super-additivity of the game and monotonic non-decreasing nature of the utilities, no group of players have the incentive to form a smaller coalition. Hence Algorithm 2 provides a solution from the core. $\square$

$$\max_{\mathcal{X}_n} \quad \sum_{m \in \mathcal{G}_1} \Big(\sum_{j \in \mathcal{M}_m} u_n^j(\mathbf{x}_n^j) - C_n^j(\mathbf{x}_n^j) \Big) \; \forall n \in \mathcal{G}_2, \tag{13a}$$

$$\text{s.t.} \quad \sum_j x_{n,k}^j \leq C'_{n,k}, \; \forall k \in \mathcal{K}, \forall j \in \mathcal{M}_m, \forall m \in \mathcal{G}_1, \tag{13b}$$

$$x_{n,k}^j \leq r_{n,k}^{'j}, \quad k \in \mathcal{K}, \forall j \in \mathcal{M}_m, \forall m \in \mathcal{G}_1, \tag{13c}$$

$$x_{n,k}^j \geq 0, \quad k \in \mathcal{K}, \forall j \in \mathcal{M}_m, \forall m \in \mathcal{G}_1, \tag{13d}$$

$$u_n^j(x_{n,k}^j) \geq C_n^j(x_{n,k}^j), \quad \forall j \in \mathcal{M}_m, k \in \mathcal{K}, n \in \mathcal{G}_2. \tag{13e}$$

Remark 3. *Algorithm 2 is a distributed algorithm, where players need to exchange information related to the request matrices, and capacity vectors. Furthermore, the information regarding the order for Step* 5 *also needs to be broadcast to all game players.*

Remark 4. *Any payoff allocation from the core is Pareto-optimal.*

As mentioned earlier, Algorithm 2 is basically a family of algorithms that can provide allocations from the core. Different variants of the algorithm arise on the basis of order in which *for-loop* in Step 5 is executed, i.e., how players in $\mathcal{G}_2$ are allowed to share their resources with players in $\mathcal{G}_1$.

Lemma 2. *The allocation decision obtained using Algorithm 2 always belong to the core irrespective of the order in which the* for-loop *of Step* 5 *is executed.*

In the next section, we evaluate the performance of our resource sharing and allocation framework.

V. Experimental Results

We evaluate the performance of proposed resource sharing and allocation framework for 3 player (domains) and 20 applications (per domain) scenario, where domain 1 has resource deficit and domains 2 and 3 have resource surplus. The communication cost between SP 1 and SP 3 is higher than between 1 and 2. Each player has three different types of resources ($K = 3$), i.e., storage, communication and computation. Without loss of generality, the model can be extended to include other type of resources/parameters. We use linear and sigmoidal utilities for all the players. The requests and capacity vectors $C_n, \forall n \in \mathcal{N}$ are randomly generated for each setting within a pre-specified range[3]. To show the advantage of resource sharing, we allocate larger capacities to certain players that share the available resources with other domains

[3]The larger the number of applications in our simulation settings, the larger is range for random number generation.

TABLE I
PLAYER PAYOFF IN DIFFERENT COALITIONS FOR A 3 PLAYER - 20 APPLICATION GAME WITH $\mu = 0.01$

Coalition	Player 1	Player 2	Player 3	Value of coalition
{1}	582.74	0.00	0.00	582.74
{2}	0	88.61	0	88.61
{3}	0	0	88.59	88.59
{1, 2}	583.74	220.01	0	803.75
{1, 3}	583.33	0	219.98	803.31
{2, 3}	0	117.45	117.45	234.90
{1, 2, 3}	644.01	188.24	136.21	968.46

and assist other players in meeting demand in order to increase their utilities. Simulations were run in `Matlab R2019b` on a `Core-i7` processor with `16 GB RAM`. To solve the optimization problems, we used the `OPTI-toolbox` [8].

A. Results for Strategy 1– Uniform Application Priority

1) Verification of game-theoretic properties: In Table I, we present results for a 3-player 20-application game with strategy 1 that verify different game theoretic properties such as individually rationality, group rationality, super additivity and show that the obtained allocation is from the core. The pay-off all players receive in the grand coalition, i.e., $\{1, 2, 3\}$ is at least as good as players 1, 2 and 3 working alone. This shows that the solution obtained using Algorithm 1 for the grand coalition is individually rational. Similarly, the value of coalition is the sum of pay-offs all players receive is the value of coalition, hence our solution is group rational. Furthermore, with an increase in the coalition size, the value of coalition also increases. Hence, the grand coalition has the largest value, which shows the super-additive nature of the game. Also, no player has any incentive to divert from the grand coalition and form a smaller coalition. Hence, the grand coalition is stable and the allocation we obtain using Algorithm 1 is from the core.

Figure 1 compares the utility, resource utilization and user satisfaction of different service providers (SPs)/domains using strategy 1 when SPs are working alone and when they are part of the grand coalition. It is evident that the utility of SPs 2 and 3 increase as it shares its resources with SP 1 whereas SP 1's utility increases due to increase in user satisfaction. Similarly, resource utilization and average request satisfaction for applications improve.

B. Results for Strategy 2– Priority for Native Applications

For strategy 2, Figure 2 compares the utility, resource utilization and user satisfaction of different service providers (SPs) when SPs are working alone and when they are part of the grand coalition. There are two different results for the grand coalition, i.e., $\{1, 2, 3\}$ (GC^1) and $\{1, 3, 2\}$ (GC^2). For GC^1, player 2 preceded player 3 in Step 5 (of Algorithm 2) while player 3 preceded player 2 in Step 5 for the grand coalition in GC^2. It is evident that utility of SPs improve in the grand coalition when compared with working alone whereas the user satisfaction of SP 1 (SP with resource deficit) also improves in the grand coalition. Furthermore, resource utilizations of SP 2 and 3 also increase as they provide their resources to SP 1. Both GC^1 and GC^2 are from the core, however, the value of coalition for GC^1 is higher than GC^2, because the cost of communication between SPs 1 and 2 are less than 1 and 3. Furthermore, comparing Figures 1 and 1, it is evident that performance of GC in Figure 1 is similar to GC^1 in Figure 2. This is because the higher communication cost between SP 1 and SP 3 forces most of SP 1's applications are provided resources by SP 2. The utility of all CSPs increased with our proposed resource sharing framework, along with resource utilization and application request satisfaction.

VI. RELATED WORK

There have been a number of solutions proposed in the literature related to resource allocation in different domains such as mobile edge clouds [9]–[17] and cloud computing. Xu et al. [12] propose a novel model for allocating resources in an edge computing setting where the allocation of distributed edge resources is decoupled from service provisioning management at the service provider side. He et al. [13] relaxes the long-held assumption that storage resources are not shareable and study the optimal allocation of both shareable and non-shareable resources in a mobile edge computing setting. Liu et al. [16] modeled the interaction among cloud service operators and edge service owners as a Stackelberg game for maximizing the utilities of both cloud and edge service providers.

Our work in this paper differs from [12]–[16] as we consider objectives of different service providers (SP) and allow resource sharing among domains. In [18], we consider the strategy where the players do not differentiate between their own applications and applications of other players, and propose a centralized algorithm that provides an allocation from the core. This work is an extension of our earlier work in [17], where we modeled the resource sharing among mobile edge clouds as a canonical cooperative game with transferable utility (TU). However, in this work, we have modeled the problem as an NTU game which is more generic than the TU game [6].

VII. CONCLUSIONS

In this paper, we have proposed a resource sharing/allocation framework based on cooperative game theory for different domains in the SDC slice. We have showed that for a monotonic non-decreasing utilities, resource sharing among domains can be modeled as convex canonical cooperative game. Therefore, the core of the game is non-empty. To address the problem of obtaining allocations from the core, two different efficient algorithms have been proposed for two allocation strategies that provide allocation from the core. Hence, the obtained solutions are Pareto optimal and the grand coalition of the domains is stable. Experimental results have showed that utilities of all service providers (domains) and user satisfaction are improved by our framework, when compared with scenarios where domains do not share resources.

REFERENCES

[1] Anton Beloglazov, Jemal Abawajy, and Rajkumar Buyya. Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing. *Future Generation Computer Systems*, 28(5):755–768, 2012.

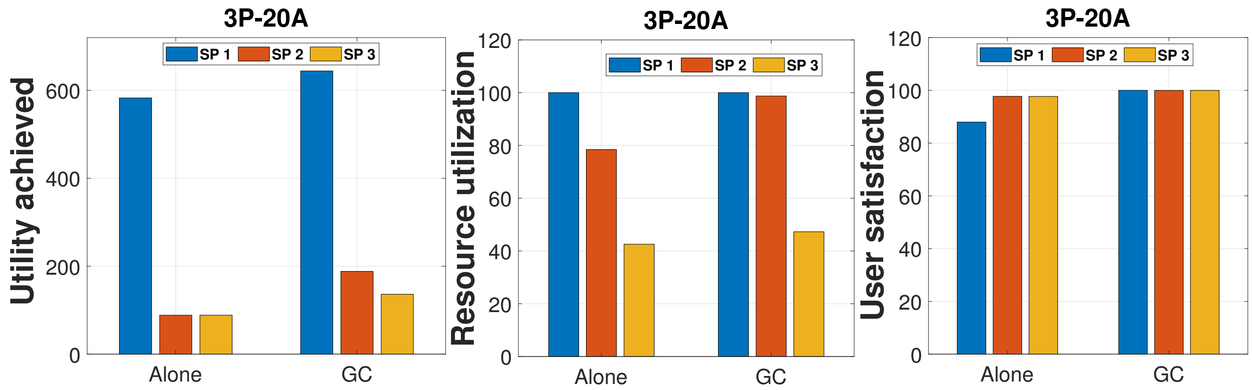

Fig. 1. Utility, resource utilization and user satisfaction for 3 player-20 application setting when domains/service providers (SPs) are working alone vs. part of coalition game for Strategy 1.

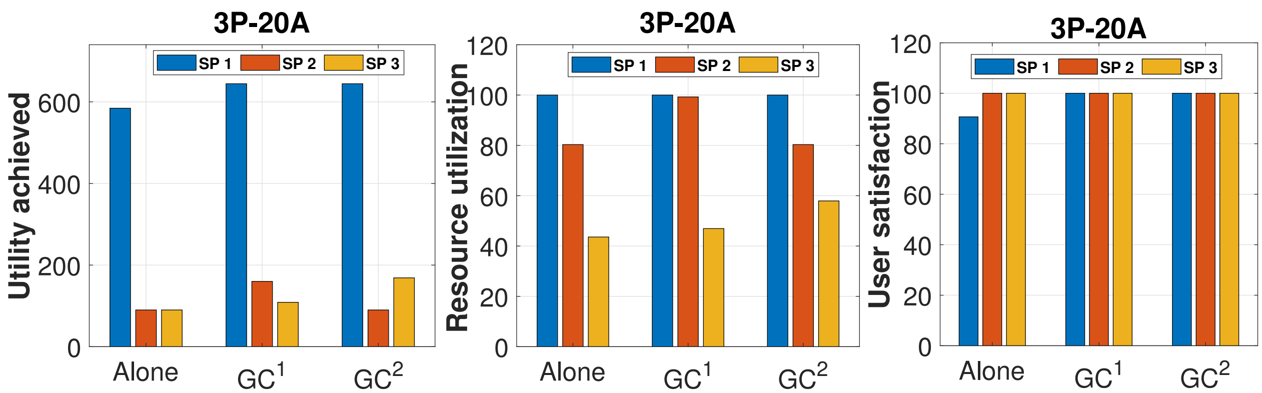

Fig. 2. Utility, resource utilization and user satisfaction for 3 player-20 application setting when domains/service providers (SPs) are working alone vs. part of coalition game for Strategy 2.

[2] Zhen Xiao, Weijia Song, and Qi Chen. Dynamic Resource Allocation using Virtual Machines for Cloud Computing Environment. *IEEE transactions on parallel and distributed systems*, 24(6):1107–1117, 2012.

[3] Mansoor Alicherry and TV Lakshman. Network Aware Resource Allocation in Distributed Clouds. In *2012 Proceedings IEEE INFOCOM*, pages 963–971. IEEE, 2012.

[4] Guiyi Wei, Athanasios V Vasilakos, Yao Zheng, and Naixue Xiong. A Game-Theoretic Method of Fair Resource Allocation for Cloud Computing Services. *The journal of supercomputing*, 54(2):252–269, 2010.

[5] Zhu Han. *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications*. Cambridge University Press, 2012.

[6] B Myerson Roger. Game Theory: Analysis of Conflict. *The President and Fellows of Harvard College, USA*, 1991.

[7] William W Sharkey. Cooperative Games with Large Cores. *International Journal of Game Theory*, 11(3-4):175–182, 1982.

[8] Jonathan Currie, David I Wilson, Nick Sahinidis, and Jose Pinto. OPTI: Lowering the Barrier between Open Source Optimizers and the Industrial MATLAB User. *Foundations of computer-aided process operations*, 24:32, 2012.

[9] Changsheng You, Kaibin Huang, Hyukjin Chae, and Byoung-Hoon Kim. Energy-Efficient Resource Allocation for Mobile-Edge Computation Offloading. *IEEE Transactions on Wireless Communications*, 16(3):1397–1411, 2017.

[10] Chenmeng Wang, Chengchao Liang, F Richard Yu, Qianbin Chen, and Lun Tang. Computation Offloading and Resource Allocation in Wireless Cellular Networks with Mobile Edge Computing. *IEEE Transactions on Wireless Communications*, 16(8):4924–4938, 2017.

[11] Stefania Sardellitti, Gesualdo Scutari, and Sergio Barbarossa. Joint Optimization of Radio and Computational Resources for Multicell Mobile-Edge Computing. *IEEE Transactions on Signal and Information Processing over Networks*, 1(2):89–103, 2015.

[12] Jinlai Xu, Balaji Palanisamy, Heiko Ludwig, and Qingyang Wang. Zenith: Utility-Aware Resource Allocation for Edge Computing. In *2017 IEEE International Conference on Edge Computing (EDGE)*, pages 47–54. IEEE, 2017.

[13] Ting He, Hana Khamfroush, Shiqiang Wang, Tom La Porta, and Sebastian Stein. It's Hard to Share: Joint Service Placement and Request Scheduling in Edge Clouds with Sharable and Non-sharable Resources. 2018. To appear in IEEE ICDCS.

[14] Huaqing Zhang, Yong Xiao, Shengrong Bu, Dusit Niyato, F Richard Yu, and Zhu Han. Computing Resource Allocation in Three-Tier IoT Fog Networks: A Joint Optimization Approach Combining Stackelberg Game and Matching. *IEEE Internet of Things Journal*, 4(5):1204–1215, 2017.

[15] Qichao Xu, Zhou Su, Qinghua Zheng, Minnan Luo, Bo Dong, and Kuan Zhang. Game Theoretical Secure Caching Scheme in Multi-Homing Edge Computing-enabled Heterogeneous Networks. *IEEE Internet of Things Journal*, 2018.

[16] Yang Liu, Changqiao Xu, Yufeng Zhan, Zhixin Liu, Jianfeng Guan, and Hongke Zhang. Incentive Mechanism for Computation Offloading using Edge Computing: A Stackelberg Game Approach. *Computer Networks*, 129:399–409, 2017.

[17] Faheem Zafari, Jian Li, Kin K Leung, Don Towsley, and Ananthram Swami. A Game-Theoretic Approach to Multi-Objective Resource Sharing and Allocation in Mobile Edge. In *Proceedings of the 2018 on Technologies for the Wireless Edge Workshop*, pages 9–13. ACM, 2018.

[18] Faheem Zafari, Kin K Leung, Don Towsley, Prithwish Basu, and Ananthram Swami. A Game-Theoretic Framework for Resource Sharing in Clouds. *arXiv preprint arXiv:1904.00820*, 2019.