Socially Aware Multi-Resource Trading for IoT Applications in Smart Cities using Auction theory

Sara Ranjbaran*, Amir Reza Jafari†, Noel Crespi†
*Aalto University, Espoo, Finland {sara.ranjbaran@aalto.fi}
†Samovar, Telecom SudParis, Institut Polytechnique de Paris, 91120 Palaiseau, France
{amir-reza.jafari_tehrani, noel.crespi}@telecom-sudparis.eu

***Abstract*—The availability of a large number of devices in the Internet of Things (IoT) allows us to carry out distributed applications in a collaborative fashion. This approach is especially beneficial in smart city scenarios, wherein different types of resources (i.e., for sensing and actuation, in addition to computation and storage) are key to providing effective location-based services. This article addresses sharing heterogeneous resources for IoT applications in smart cities. Specifically, it takes a game-theoretic approach and addresses the problem of allocating resources from different devices as a combinatorial double auction. Features specific to IoT devices, including their affinity (e.g., ownership and co-location), are leveraged to form groups that offer resource bundles with a certain reliability. The proposed solution incurs a low computational complexity and achieves several important economic properties: incentive compatibility, individual rationality, and a balanced budget. Simulations based on a large-scale dataset demonstrate that the combinatorial double auction is effective and can be completed in a short time.**

***Index Terms*—Resource sharing, combinatorial auction, double auction, Internet of Things, fog computing, smart cities.**

I. Introduction

The Internet of Things (IoT) is composed of a large number of interconnected devices that are able to perform different actions to support diverse applications [1]. Wherein, from one hand, many of these devices are resource constraint and cannot support fast and accurate execution of their tasks, and on the other hand, we are witnessing an ever increasing advanced handheld devices, with enhanced storage and computing capabilities where they can not only satisfy the needs of their owners but also act as a host to provide related services to users nearby. Such actions do not only involve processing and communication but also sensing, actuation and data storage. In this respect, IoT devices are very heterogeneous in terms of the number and types of their resources. For instance, a surveillance camera has some storage and processing capabilities, to annotate video streams; a smart light includes both a proximity sensor (to detect the presence of people or vehicles nearby) and an actual actuation component (to switch the light on and off accordingly).

Emerging edge computing provides an effective solution to overcome the limitations of the cloud. Indeed, different from cloud computing, edge computing is a type of decentralized computing paradigm, which moves data, computation, and storage from the data center to edge nodes of the network [2]. Edge technologies also provide more intelligent analysis and processing services near the data sources, e.g., user equipment's, intelligent vehicles, etc. In this case, the communication delay can be significantly reduced, the system efficiency can be effectively increased.

However, the existing literature has primarily focused on several challenges in edge computing (i.e., security and reliability, heterogeneity). Instead, economic aspects in the reliable allocation of heterogeneous IoT resources have received limited attention (Section II). This article leverage this concept to address the above-mentioned issues by taking a game-theoretic approach to trade resources among different IoT applications. Specifically, we adopt auction theory to provide fair resource allocation between buyers, and sellers [1] in the case of competition. Indeed, as the interest of heterogeneous devices is usually inconsistent, so we opt double auction mechanism to consider the interests of all parties [3]. However, different from the existing edge-related auction which only considers the price, some non-price attributes (location, affinity, and computing power) are also considered for providing fair resource allocation in our proposed method. In real scenarios, due to the different type of tasks and different capabilities of IoT devices, both buyers and sellers may have different preferences over each other (i.e., a device prefer to choose a seller with higher capabilities when processing safety-related tasks). This mechanism takes into account the locality characteristics of the systems, where mobile devices can only offload tasks to the edge node in

[1] We use buyer and seller terms referring to IoT devices who is need of services and the ones with spare resources to offer respectively

their proximity, and the edge node only serves the neighbouring devices with available resources. The main contributions of this work are as follows: First, it models resource sharing as a combinatorial auction in which IoT service providers and devices simultaneously place bids for a collection of heterogeneous resources. The proposed approach leverages affinity between devices (e.g., in terms of co-location and ownership) and an efficient representation of resource bundles (Section III). Moreover, it introduces a double auction algorithm that has a low time complexity and achieves several economic properties: incentive compatibility, individual rationality, and a balanced budget (Section IV). Finally, extensive simulations based on a large-scale dataset demonstrate that the combinatorial double auction is effective, as it efficiently allocates resources and is carried out within a short time (Section V).

II. Related work

Resources sharing among IoT devices has received considerable attention in the literature. For instance, Kortoçi et al. [4] focused on collaborative data storage in the opportunistic IoT, wherein mobile data collectors infrequently visit nodes. Ranjbaran et al. [5] leveraged matching theory to obtain a many-to-many assignment of resources involving heterogeneous devices.

Several works took a game-theoretic approach in characterizing the IoT and edge scenarios. Farris at al. [6] introduced the concept of the federation for mobile IoT clouds at the network edge. Specifically, the authors considered a coalition formation game and a Nash-stable solution to manage the resulting federations. Moreover, several existing works have recently adopted auction theory to solve the resource management at the edge of network [7]–[10]. It is inspired by the existing works that different types of auction methods are suitable for different types of problems in specific application scenarios. For example, a multi-round-sealed sequential combinatorial auction mechanism is used in a scenario for purchasing the combinational resources to obtain mobile edge computing service by service providers [11]. Also for fog providers, Anjan et al. [12] investigated that the combinatorial auction-based mechanism can improve the allocation of resources to create higher income. Furthermore, combinatorial clock auction is also used for live video streaming in mobile edge to improve the quality of experience of cloud services [13], [14]. Zavodovski et al. [9] designed a truthful decentralized double auction for for computational resource sharing, by explicitly targeting distributed ledger models. Kiani and Ansari [15] developed a one-way auction mechanism for resource allocation in edge computing systems which, however, does not guarantee trustfulness. Ma et al. [10] proposed a truthful double auction mechanism for resource allocation for the industrial IoT. Even though the approach pursued here is similar, this work specifically addresses heterogeneous IoT resources, resulting in a combinatorial auction, where, our proposed mechanism allows one seller to serve multiple buyers simultaneously. Moreover, different from the existing edge-related auction which only considers the price, some non-price attributes (location, affinity, and computing power) are also considered for providing fair resource allocation in the system.

III. Resource Trading

This section first introduces the system model and then formulates the resource-sharing problem, followed by an auction design that reduces its complexity. It finally introduces the matching principle to allocate resources to devices as well as the actual payment mechanism.

A. System Model

The reference scenario is represented by a smart city neighbourhood, consisting of N public and private entities, $\mathcal{E} = \{E_1, E_2, \ldots, E_N\}$, each owning a set of $\mathcal{D}_{E_i}$ devices $\mathcal{D}_{E_i} = \{d_j : d_j \in E_i\}$. These devices are heterogeneous, in the sense that they offer different types of services (equivalently, resources). Specifically, the set of the offered services by device d_i is $\mathcal{S}_{d_i} = \{S^\tau : \tau \in \mathcal{T}\}$, where τ is the type available services in the set $\mathcal{T}$; these could be sensing, actuation, computation, storage and bandwidth resources for instance. Devices can be mobile: they alternate between moving from one location to another and staying at a certain location for some time, which is assumed to be known in advance or estimated through machine learning methods [16].

Devices leverage a nearby edge node (EN) for management purposes: they inform the EN about their capabilities as well as users' preferences and update their information as their location changes.

Devices act as Micro-Providers (MPs) for services, in the sense that they can offer part or all of the resources needed for user requests. Assuming that $\mathcal{R}$ is the set of all the requests in the system, the individual request r submitted by client i and valid until time t_i is described by:

$$r_i \triangleq \; < t_i, \mathcal{P}_{(i)}, b_i, t_i^s, t_i^e, q_i, \phi_{(i)}, f_i, l_i > \qquad (1)$$

Here, $\mathcal{P}_{(i)} = (\rho_{i1}, \ldots, \rho_{ij}, \ldots \rho_{ik})$ with $\mathcal{P}_{(i)} \in Z^k : \rho_{ij}$ is the amount of resource with type $j \in \mathcal{T}$ requested ($\rho_{ij} > 0$). The client makes a *bid* $b_i \in R^+$ representing the maximum price that it is willing to pay for the whole bundle $\mathcal{P}_{(i)}$. Note that this is a reserved price based on a private monetary valuation for the requested services. The terms t_i^s and t_i^e, indicate the period (in terms of starting and ending time) during which the resources are needed, while q_i specifies the maximum number of devices that can be used to fulfill the request. This is an important parameter, since the higher the number of allowed devices, the simpler is to find matching devices, but it is also may endanger the reliability of service as there is a higher possibility that one of the assigned devices acts non-cooperative (e.g. leave the coverage area of the edge node, thus delaying the service provisioning. Moreover, $f_i \in \{0,1\}$ is a bundle preference indicator: $f_i = 1$ means that the client needs either the whole bundle or no services at all, whereas $f_i = 0$ signifies that any subset of the requested service types (between 0 to k) is enough. Moreover, $\phi(i) \in [0,1]$ denotes the importance level of the requested service: $\phi_i = 1$ indicates that the service is extremely valuable (i.e. the client would like to get the service from a trusted micro-provider with higher capabilities when processing safety-related tasks). whereas $\phi_i = 0$ means that the client is willing to obtain resources with a focus on price, without any preferences on the bundle. Finally, the request is tagged with the location l_i of the client, broadly defined (for instance, as either GPS coordinates or a network address). As this mechanism takes into account the locality characteristics of the IoT systems, where clients only offload tasks to ENs in the proximity and ENs only serve clients with their neighbouring MPs.

Moreover, MPs may submit their *offers* to the EN announcing their owned spare resources, overall indicated as the set $\mathcal{O}$. The offer from MP j is described similarly to the requests:

$$o_j \triangleq < t_i, \mathcal{P}_{(j)}, t^s_j, t^e_j, C_j, q_j, l_j > \qquad (2)$$

where $\mathcal{P}_{(j)} = (\rho_{j1}, ..., \rho_{jl})$ and ρ_{jl} is the amount of the resource of type $l \in \mathcal{T}$, whereas t^s_j and t^e_j are the start and ending times resources are available. Here, $C_j = (c_{j1}, ...c_{jl})$ describes the costs associated with each type of resource. These values are considered a reserved price to ensure a minimum amount of resources so that the micro-provider does not suffer losses from resource sharing and allocation. Finally, q_j is a quota in terms of the maximum number of requests that a device can be matched with and l_j is the device location. The bidding process takes place over time, which is supposed to be divided into T_s timeslots with a duration equal to T (seconds). Devices with spare resources and clients needing services submit their offers and requests (respectively) to the EN at each timeslot t.

B. Problem Formulation

In this paper, we take forward our research by proposing a double combinatorial auction-based mechanism for the resource-sharing problem. From the perspective of mechanism design, optimization targets for economic performance can be divided into two main aspects. One is revenue maximization, which pursues the benefit of the seller. While the other one is the efficiency mechanism, which focuses on maximizing social welfare. For a double auction, welfare is the difference between the total value of the buyers and total cost of the sellers [17]. So, as we want to motivate clients and providers equally, we focus on maximizing social welfare. Indeed, social welfare is equivalent to allocating resources to the buyers who value them most, buying those goods from the sellers which have the lowest prices.

Recall that each client i declares a bid $b_i > 0$ and has a private true valuation v_i for any combination of resources expressed according to Eq. (1). The same condition is also established for available resources; in other words, the micro-provider j may have more than one free resource at a given time with the associated cost C_j for all resources that have been specified in its offer. Indeed, as each EN is considered independently, so without loss of generality we can formulate the problem as welfare-maximization for each EN. We define the the allocation matrix $\mathcal{X}_\epsilon = [x^\epsilon_{ij}]$ which represent the allocation between request $i \in \mathcal{R}^\epsilon$ and offer $j \in \mathcal{O}^\epsilon$. Which $x^\epsilon_{ij} = 1$ states the best possible matching the given scenario when request i is assigned to offer j.

We define the problem of welfare-maximization as follows:

$$\max \sum_{i \in \mathcal{R}^\epsilon} \sum_{j \in \mathcal{O}^\epsilon} v_i x^\epsilon_{ij} - \sum_{i \in \mathcal{R}^\epsilon} \sum_{j \in \mathcal{O}^\epsilon} x^\epsilon_{ij} \psi^\epsilon_{ij} C_j \qquad (3)$$

subject to

$$\sum_{j \in \mathcal{O}^\epsilon} \leq q_i, \;\; \forall i \in \mathcal{R}^\epsilon \qquad (4)$$

$$\sum_{i \in \mathcal{R}^\epsilon} \psi^\epsilon_{ij} x^\epsilon_{ij} \leq \rho_j \qquad (5)$$

$$\rho_{i,k} x^\epsilon_{ij} \leq \rho_{jk}, \;\; \forall i \in \mathcal{R}^\epsilon, \forall j \in \mathcal{O}^\epsilon, \forall k \in \mathcal{T}, \qquad (6)$$

$$v_i \geq \psi^\epsilon_{ij} C_j, \qquad (7)$$

$$v_i \geq 0, \;\; \forall i \in \mathcal{R}^\epsilon, \qquad (8)$$

$$C_j \geq 0, \;\; \forall i \in \mathcal{O}^\epsilon, \qquad (9)$$

$$x^\epsilon_{ij} \in \{0,1\} \qquad (10)$$

The objective function, Eq(3), is aiming at maximizing the welfare for all requests and offers accepted in under EN ϵ. Where, ψ^ϵ_{ij} is the resource fraction of offer j allocated to request i. Eq. (4) ensures that a request is matched to at most q_i devices. Eq. (5) establishes the feasibility of resource allocation, i.e., that there are sufficient resources to serve the requests. Eq. (6) ensures that if offer j is assigned to request i, j has sufficient quantity of resource type k to serve i. Eq. (7), ensures that the valuation of the client is greater than the cost of allocated resources. Eq. (9) and Eq. (10) state that valuation and cost can only be non-negative rational numbers. Finally, Eq. (10) indicates that the decision variables are binary integers.

C. Auction Design

To address the complexity of the problem introduced above, the following presents an auction-based mechanism involving a set of agents (clients) and a set of micro-providers (devices). The process takes place over multiple rounds; for simplicity, the rest of the discussion focuses on a single round. The EN act as the broker for the auction.

The original problem is relaxed first by restricting the permitted combinations to a cluster of available resources, determined according to their characteristics and their affinity. Specifically, each cluster is represented by a subtree with a given level of importance. Assume that $k \subseteq A$ represents a combination of items and $\mathcal{K}$ is the set of all combinations $|\mathcal{K}| = 2^n$. Accordingly, the set $\mathcal{C}$ of partitions can be defined as:

$$\mathcal{C} = \{\omega \subseteq \mathcal{K} | k, k' \in \omega \Rightarrow k \cap k' = \emptyset\} \tag{11}$$

where partition ω is the set of pairwise disjoint subsets of items. In detail, a directed tree structure $\mathbb{T}$ is created for $\mathcal{C}$ as follows: the vertices correspond to the permitted combinations; and (k, k') is an edge in $\mathbb{T}(\mathcal{C})$ if and only if k covers k' in $\mathcal{C}$ (i.e., there is a direct path from k to k'). Formally, (k, k') is an edge if there is no subset $k'' \in \mathcal{C}$ such that $k \supset k'' \supset k'$; that is, k' is the tail and k is the head of the edge [18]. Moreover, every k has at most one incoming edge in the directed tree. Note that a set of permitted combinations $\mathcal{C} = \{\omega_i\} \quad i \in \{1, \dots, d\}$, where d is the depth of the tree, form a tree structure if for any pair of subsets $\forall k, k' \in \mathcal{C}$ are disjoint (i.e. $s \cap s' = \emptyset$) or one is the subset of the other. In the considered setting, each level in the tree indeed forms a partition (i.e., every single item is just included in at most one subset) such that the number of single items in each level is the same (i.e., $\forall i \quad |\omega_i| = |\omega_{i+1}| \quad i \in \{1, 2, \dots, d\}$) where, ω_i denote the partition in level i. Also, for each edge $(k, k') \in \mathbb{T}(\mathcal{C})$ the combination k is larger than k' ($|k| > |k'|$) if k covers k'. The leaves of the tree $k \in \mathcal{C}$ correspond to single resources ($k \in \omega_1$ and $|k| = 1$). At each level, partitions are formed in such a way that the importance among the items in a given subset is higher than a threshold of Δ. Indeed, each subset in ω_i at level i form a subtree that can be considered independently. Such a partitioning scheme allows the assignment of the best possible matching according to user preferences. The cost of each combination is considered superadditive: $C(k \cup k') = C(k) + C(k') + \mu(C(k) + C(k'))$ where $0 < \mu < 1$, where this coefficient is larger in the first tree ($\mu(\mathbb{T}_1) > \mu(\mathbb{T}_2)$), as more appealing combinations are assumed to be more expensive. An upper bound g for the size of permitted combinations is also defined, as the maximum number of physical devices that can be assigned to a given request. Consequently, the depth of the tree is limited by this parameter ($d = g + 1$), thereby reducing computational demands. It is worth mentioning that the permitted combinations are temporally compatible, i.e. they are all available in a certain period of time.

TABLE I: Affinity types [20] and corresponding values.

Type	**Abbr.**	F
Owner Object Relationship	OOR	1.0
Co-Location Object Relationship	CLOR	0.8
Co-Work Object Relationship	CWOR	0.8
Social Object Relationship	SOR	0.6
Parental Object Relationship	POR	0.5

D. Matching Principle

After collecting all requests and the set of allowed combinations, it remains to derive the actual mappings according to certain principles. Note that the characteristics of available resources are not public in the bidding phase, so clients cannot explicitly target them. Therefore, a matching principle is needed to find the most suitable combination for each request. This allows the winner request to obtain the best approximation of the services specified in the bids.

For this purpose, the concept of *reliability* among devices is considered on the basis of two main factors. The first is device *affinity*(i.e., object sociality), expressed as the coefficient $F_{i,j}$ that represents the type of relationship between IoT devices i and j according to the social IoT [19], (as indicated in Table I). The second factor is the *reputation* of a device, based on the history of successful transactions carried out in the past. Specifically, $W_i = \frac{k_i}{K_i}$ is the transaction success rate for device i, where k_i is the number of successful transactions involving device i and K the total number of transactions. Accordingly, the reliability involving devices i and j is:

$$R_{ij} = \gamma A_{ij} + (1 - \gamma) W_i \tag{12}$$

where γ is a weight determined as a trade-off between the impact of the two terms.

While reliability is important, it is not the only factor having an impact on how clients are willing to accept resources – the cost is also significant, especially for low values of preference (e.g., $\phi_{(i)} < 0.5$). Accordingly, a matching quality level between request r and the feasible combination a is defined as:

$$\mathcal{S}(r, a) = \sum_{i=1}^{\mathcal{T}_a \cap \mathcal{T}_r} \left\lceil {\rho'}_a^i - {\rho'}_r^i \right\rceil \left(\frac{{\rho'}_a^i}{({\rho'}_a^i - {\rho'}_r^i)^2 + 1} \right) + \phi_{(r)}(R_{ir}) \tag{13}$$

Here, a is a permitted combination that the request r_i can afford ($b_i > c_a$), whereas ${\rho'}_a^i \in [0,1]$ and ${\rho'}_r^i \in [0,1]$ are normalized value of resource type i in subset a and request r respectively, calculated based on the maximum amount of available resources in the market. Furthermore, R_{ir} indicates the reliability of request i for the device owning resource j in subset a. Eq. (13) allows to rank the subsets for a particular request with at least one common resource during the same time period, namely, $|\mathcal{T}_a \cap \mathcal{T}_r| > 1$. For the case of $f_i = 1$ (the client requests the whole bundle), the subset needs to satisfy $|\mathcal{T}_a \cap \mathcal{T}_r| = |\mathcal{T}_r|$.

E. Payment Mechanism

The actual payment mechanism relies on the concepts of utility and pricing based on critical value [21]. The utility of client i from an accepted request is r_i as $u_i = v_i - p_i$, where v_i is the private valuation for the requested bundle. Similarly, the utility for micro-provider j offering some resource is $u_j = \pi_j - c_j$, where π_j is the revenue of the provider from sharing that resource. By definition, the utility is zero if the participant in the auction is not assigned a resource. The critical value v_i^c for request r_i is a unique value that a client must declare to win the requested bundle. In other words, for any $v_i > v_i^c$ the request wins and pays its critical value, while for any $v_i < v_i^c$ the client loses the auction and pays zero. The critical value is computed based on the second loser request r_z who would win if r_i did not participate. Such a payment method is monotone as increasing the bid b_i for a smaller combination does not result in a client losing. Furthermore, declaring a bid below the true valuation may lead to a loss in the mechanism. On the other hand, offering less for more items does not incur a lost bid to win.

IV. Resource Sharing through Combinatorial Auction

This section presents the proof of the economic properties for proposed heterogeneous REsource Sharing through Combinatorial Auction (RESCA). RESCA is a double auction mechanism where both micro-providers and clients submit their offers and requests to the EN by specifying their costs and valuations (respectively). It covers both winner determination (equivalently, finding the best match for devices allocated to tasks) and the actual payment. RESCA is executed at the EN – acting as the broker – during each round l of the auction; it guarantees that none of the devices are allocated if they are below their declared cost, and also no client pays more than what they have offered in their requests.

A. Analysis of RESCA

The following proves that RESCA satisfies the following economic properties.

- *Computational efficiency (CE)*: the trading algorithm should be computed in polynomial time.
- *Incentive compatibility (IC) or Truthfulness*: the dominant strategy for the SPs and each device is to bid their true valuation, otherwise their profit reduces.
- *Individual rationality (IR)*: both the SPs and devices obtain non-negative profit from the auction.
- *Balanced budget (BB)*: the total payment of devices should exceed the total charging of the SPs.

The rest of the section proves each of these properties; the proofs refer to a single round of the auction since the pricing policy of a given round is independent of the others.

Theorem 1 (Computational efficiency)**.** *RESCA has a polynomial complexity.*

Proof. The main task of the RESCA mechanism is to match the requests with a subset of offers. Accordingly, the complexity of winner determination in the tree structure in $O(|\mathcal{C}|(|\mathcal{C}| + n))$, wherein $|\mathcal{C}|$ is the number of all permitted combinations. For any directed tree structure, it is $|\mathcal{C}| < 2n - 1$ [22]. Therefore, the breadth-first search finds a match for each request in the directed tree $\mathbb{T}(\mathcal{C})$, with $|V| = 2n - 1$ and $|E| = 2n - 2$, in $O(n^2)$ time, Moreover, the mechanism prunes the tree at each search node, leading to a speedup in the later phases. Indeed, each subtree corresponds to an independent subproblem, namely, the subset of the combinations and the associated requests. The worst case is represented by all clients bidding exactly one unit for a single type of resource. The complexity of ranking the bids is $O(m \log m)$, where m is the number of requests. Finally, the time to carry out the payment is $O(m)$, as it involves iterating over the set of winner clients and the bids are already sorted [23]. So the overall complexity of the algorithm is $O(mn^2)$. □

Theorem 2 (Incentive compatibility)**.** *RESCA is incentive-compatible (i.e., truthful).*

Proof. Recall that the payment method discussed in the previous section is monotonic and the price is decided based on the critical value, which are necessary conditions for incentive compatibility [24]. Specifically, a truthful request r_i wins the combination s_i by biding the value b_i for which it pays the critical value v_i^c, namely, $v_i^c \leq b_i$. There are two possible untruthful behaviours of the participants.

- If client i overbids, it is $b_i > v_i$ and the request i still can win the auction but its payment does not change, as it corresponds to its critical value $v_i^c < v_i$.
- If client i underbids, it is $b_i < v_i$ and the client may still win the auction but its payment does not change, unless $p_i > b_i$ in which case its utility is zero as the client loses the auction.

The same reasoning can be used for the offers of micro-providers. As a result, there is no incentive for participants to bid untruthfully, thus RESCA is incentive-compatible. □

Theorem 3 (Individual rationality)**.** *RESCA satisfies individual rationality.*

Proof. The utility of client i is zero if request r_i does not win the auction. Similarly, the utility of a micro-provider is zero if it cannot trade any resource. If a micro-provider trades a resource it is paid no less than its reserved price. Moreover, if a request wins the auction, it pays less than its own bid, according to critical value payment. Consequently, both participants have a non-negative utility by participating in the auction, which establishes the individual rationality satisfied by RESCA. □

Theorem 4 (Balanced budget)**.** *RESCA maintains a balanced budget.*

Proof. Given any reserved price c_k for combination k, the bid of the winning device j is b_j, with $b_j > c_k$. In addition, the final payment for the winning request is based on a critical pricing model where $p_j > c_k$. In other words, the broker does not pay extra money in the auction, thereby resulting in a balanced budget. □

V. Performance Evaluation

This section evaluates the performance of the proposed resource-sharing mechanism for IoT applications in a smart city scenario. It first introduces the considered setup and the evaluation methodology, then presents the obtained results.

A. Setup and Methodology

The evaluation is conducted by using a custom simulator written in Python. The considered scenario includes IoT devices acting as micro-providers for four types of resources: sensing, computation, storage, and actuation. IoT devices are connected to the Internet and are within a certain geographical area served by an EN. The specific features of the devices are selected from the IoT network dataset[2] in [25]. Specifically, the dataset encompasses more than 16,000 devices, both public and private; it also provides information about IoT applications and services deployed in a metropolitan area as well as the social relations between the devices, including mobility. The network includes an EN serving 1,000 devices with different capabilities, selected according to the values in [5]. The EN periodically collects the available resources from the devices, after which the auction takes place with RESCA.

[2]http://social-iot.org/index.php?p=downloads

TABLE II: Simulation parameters.

Parameter	Value
Number of users	200
Number of devices per user / municipality	5
Number of available resources	[1,000, 4,000]
Sensing units per service request	[1, 8]
Computation units per service request	[1, 15] MFLOPS
Storage units per service request	[10, 1,500] MB
Actuating units per service request	[1, 3]
Number of sensing resources per device	[2, 8]
Number of actuation resources per device	[1, 3]
Amount of RAM per device	[0.1, 16] GB
Amount of storage per device	[0.1, 1,024] GB

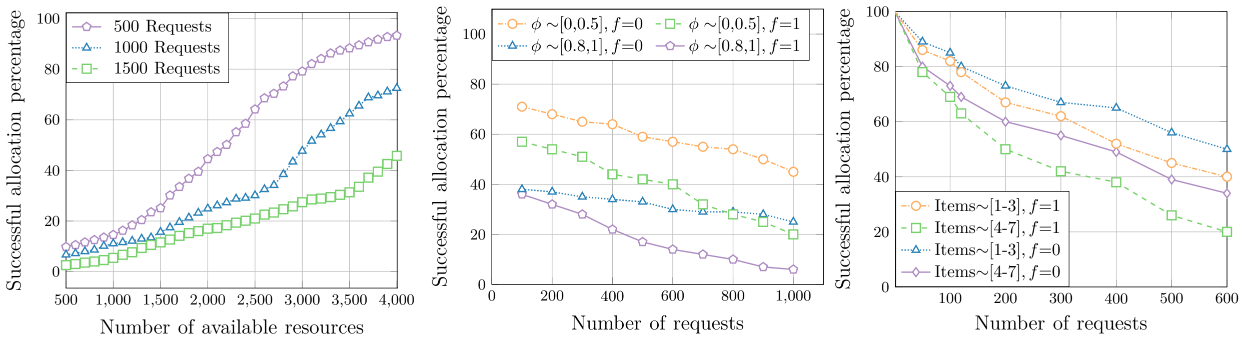
Simulations are carried out by using the independent replication method with 40 iterations for each experiment. The results report the average values; the corresponding standard deviations are also reported as error bars when noticeable. Table II details the parameters used in the simulation.

B. Simulation Results

The rest of this section presents the simulation results by dividing them into two different categories. The first one characterizes resource allocation, primarily in terms of the fraction of resources allocated successfully with respect to the corresponding requests (Figure 1). The second set of experiments focuses on the economic properties of RESCA in terms of revenue and prices (payments) as well as the runtime of the winner determination part of the algorithm (Figure 2).

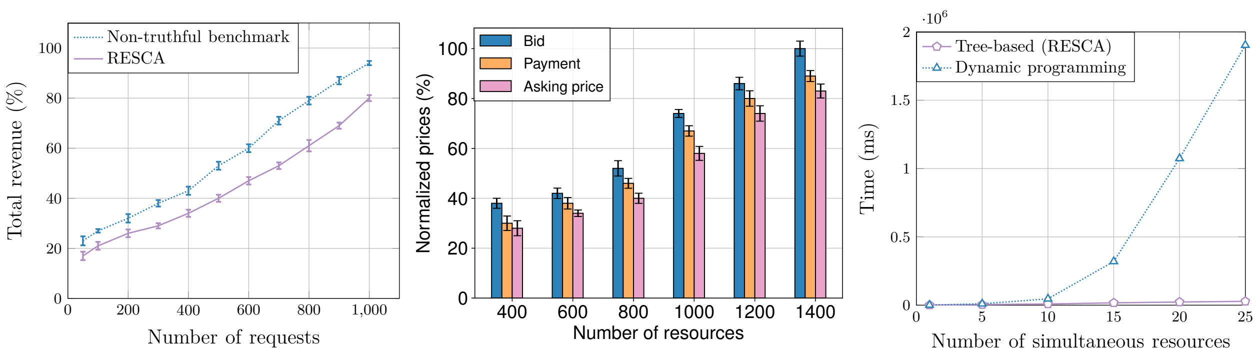
Fig. 1a reports the percentage of successfully allocated resources as a function of the available resources for different amounts of requests. Clearly, the chances of success increase with the availability of resources in all cases. Initially, the success rate increases almost linearly, while it starts growing faster after a value of available resources that depends on the number of requests. In any case, RESCA successfully allocates more than 70% of the resources when there are up to 1,000 requests, demonstrating its sharing efficiency.

Fig. 1b also shows the percentage of successfully allocated resources, but this time as a function of the number of requests, for different values of importance (i.e., ϕ) and bundle preference (i.e., f). The simulations consider the number of items per request in the range [1-6] and 1,000 available resources. Of course, the fraction of successful allocations decreases as the number of requests increases, but the impact of the two different parameters is more significant. When the importance factor is relaxed (i.e., the affinity is not considered when $\phi = 0$) the success rate increases. In turn, the success rate decreases when the affinity is high (i.e., reliability is the main factor for clients). This happens since RESCA selects resources with OOR affinity to maximize the importance preference of clients. The bundle preference



(a) (b) (c)

Fig. 1: (a) Percentage of successful assignments, (b) impact of different preferences on successful assignment. (c) the effect of requests size on successful assignment for 1500 available resources, for two different scenarios



(a) (b) (c)

Fig. 2: (a) Payment and bidding price of clients; (b) payment and asking prices for 100 winning requests. (c) Running time in comparison with dynamic programming.

parameter also affects the success rate significantly, in which the chance of winning resources increases for devices that are more flexible in their requests (i.e., $f = 0$).

Moreover, Fig. 1c shows the percentage of successfully allocated resources as a function of the number of requests, for different distributions of items per request. Here, a fixed level of importance $\phi = 0.7$ is considered. Also here it is apparent how the percentage of success increases for flexible requests, compared to those with stricter requirements (i.e., a client willing to always get a requested resource entirely). The same happens when fewer items are requested.

Economic properties Figure 2a shows the total revenue of RESCA compared to a non-truthful benchmark as a function of the number of requests. The benchmark is a double auction using the same algorithm as in RESCA, but with a payment mechanism that is not based on the critical value. Clearly, the non-truthful benchmark achieves a higher revenue than RESCA. However, the difference with RESCA is relatively small, especially when there are fewer than 500 requests. This result is remarkable, as RESCA also has the potential to enable efficient use of spare resources in the long term (i.e., over multiple rounds of the auction).

Moreover, Figure 2b shows the total payments of the winning clients together with their bidding prices and the asking price of selected micro-providers. In this case, the number of requests is set to 300, and the number of resources is varied between 400 to 1,400; all values are normalized in a range of (0-100] for clarity. The figure clearly shows that the winning clients acquire their requested bundle yet their payments do not exceed their valuations. At the same time, the selected micro-providers earn more than their reserved price, which motivates them to participate in the next rounds of resource trading. Both these results confirm the analysis in Section IV-A.

Finally, Figure 2c shows the running time for the tree-based winner determination in RESCA as well as the same for the optimal algorithm in [26], based on dynamic programming. The experiments were carried out on a machine with a 3.2 GHz ARMv8-A processor with 8 cores and 16 GB of RAM. The figure clearly shows how the computation time of RESCA is always below 30 seconds and significantly lower than that of the solution employing dynamic programming. The latter obtains extremely high running time when the number of simultaneous resources is 20 or more (more than 15 minutes). This confirms that winner determination algorithms employing dynamic programming are not scalable, whereas the solution in RESCA is fast irrespective of the number of resources.

VI. Conclusion

This article has considered an IoT scenario in which devices share heterogeneous resources according to the fog networking paradigm. In particular, it has characterized the value of resource combinations by means of the reliability of the corresponding devices to provide services relying on these resources. Moreover, it has been considered an efficient representation of resource bundles and a combinatorial double auction mechanism to assign these resources to different services. The resulting mechanism for resource sharing has been analytically characterized in terms of time complexity and economic properties. Finally, these have been validated by an extensive simulation study leveraging a large-scale IoT dataset, which has also characterized the efficiency of resource allocation. The obtained results have established that the proposed approach is effective in sharing resources, by also obtaining a solution in a short time. In future work, we plan to consider distributed long-term interactions between IoT devices without EN intervention. The other interesting thread is to evaluate how the behaviour of unreliable or malicious users affects the dynamics of sharing.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE internet of things journal*, vol. 3, no. 5, pp. 637–646, 2016.

[3] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE internet of things journal*, vol. 3, no. 6, pp. 1171–1181, 2016.

[4] P. Kortoçi, L. Zheng, C. Joe-Wong, M. Di Francesco, and M. Chiang, "Fog-based data offloading in urban IoT scenarios," in *The 38th IEEE International Conference on Computer Communications (INFOCOM 2019)*, May 2019, pp. 784–792.

[5] S. Ranjbaran, M. H. Manshaei, and M. Nitti, "A social-aware approach for federated iot-mobile cloud using matching theory," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. IEEE, 2019, pp. 554–559.

[6] I. Farris, L. Militano, M. Nitti, L. Atzori, and A. Iera, "Mifaas: A mobile-iot-federation-as-a-service model for dynamic cooperation of iot cloud providers," *Future Generation Computer Systems*, vol. 70, pp. 126–137, 2017.

[7] Y. Luo, W. Ding, and B. Zhang, "Optimization of task scheduling and dynamic service strategy for multi-uav-enabled mobile-edge computing system," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 3, pp. 970–984, 2021.

[8] C. Kai, H. Zhou, Y. Yi, and W. Huang, "Collaborative cloud-edge-end task offloading in mobile-edge computing networks with limited communication capability," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 2, pp. 624–634, 2020.

[9] A. Zavodovski, S. Bayhan, N. Mohan, P. Zhou, W. Wong, and J. Kangasharju, "Decloud: Truthful decentralized double auction for edge clouds," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 2157–2167.

[10] L. Ma, X. Wang, X. Wang, L. Wang, Y. Shi, and M. Huang, "Tcda: Truthful combinatorial double auctions for mobile edge computing in industrial internet of things," *IEEE Transactions on Mobile Computing*, 2021.

[11] H. Zhang, F. Guo, H. Ji, and C. Zhu, "Combinational auction-based service provider selection in mobile edge computing networks," *IEEE Access*, vol. 5, pp. 13 455–13 464, 2017.

[12] A. Bandyopadhyay, T. S. Roy, V. Sarkar, and S. Mallik, "Combinatorial auction-based fog service allocation mechanism for iot applications," in *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2020, pp. 518–524.

[13] Y.-H. Hung, C.-Y. Wang, and R.-H. Hwang, "Combinatorial clock auction for live video streaming in mobile edge computing," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2018, pp. 196–201.

[14] ——, "Optimizing social welfare of live video streaming services in mobile edge computing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 4, pp. 922–934, 2019.

[15] A. Kiani and N. Ansari, "Toward hierarchical mobile edge computing: An auction-based profit maximization approach," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2082–2091, 2017.

[16] A. Mei and J. Stefa, "Swim: A simple model to generate small mobile worlds," in *IEEE INFOCOM 2009*. IEEE, 2009, pp. 2106–2113.

[17] P. Dütting, I. Talgam-Cohen, and T. Roughgarden, "Modularity and greed in double auctions," *Games and Economic Behavior*, vol. 105, pp. 59–83, 2017.

[18] J. A. Bondy, U. S. R. Murty *et al.*, *Graph theory with applications*. Macmillan London, 1976, vol. 290.

[19] M. Nitti, R. Girau, and L. Atzori, "Trustworthiness management in the social internet of things," *IEEE Transactions on knowledge and data engineering*, vol. 26, no. 5, pp. 1253–1266, 2014.

[20] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization," *Computer networks*, vol. 56, no. 16, pp. 3594–3608, 2012.

[21] D. Lehmann, L. I. Oćallaghan, and Y. Shoham, "Truth revelation in approximately efficient combinatorial auctions," *Journal of the ACM (JACM)*, vol. 49, no. 5, pp. 577–602, 2002.

[22] M. H. Rothkopf, A. Pekeč, and R. M. Harstad, "Computationally manageable combinational auctions," *Management science*, vol. 44, no. 8, pp. 1131–1147, 1998.

[23] S. Zaman and D. Grosu, "A combinatorial auction-based mechanism for dynamic vm provisioning and allocation in clouds," *IEEE Transactions on Cloud Computing*, vol. 1, no. 2, pp. 129–141, 2013.

[24] A. Mu'Alem and N. Nisan, "Truthful approximation mechanisms for restricted combinatorial auctions," *Games and Economic Behavior*, vol. 64, no. 2, pp. 612–631, 2008.

[25] C. Marche, L. Atzori, and M. Nitti, "A dataset for performance analysis of the social internet of things," 2018.

[26] T. Sandholm, "Algorithm for optimal winner determination in combinatorial auctions," *Artificial intelligence*, vol. 135, no. 1-2, pp. 1–54, 2002.