AMACoT: A Marketplace Architecture for Trading Cloud of Things Resources

Ahmed Salim Alrawahi, *Member, IEEE,* Kevin Lee, *Senior Member, IEEE,*
and Ahmad Lotfi, *Senior Member, IEEE*

Abstract—**Cloud of Things (CoT) is increasingly viewed as a paradigm that can satisfy the diverse requirements of emerging IoT applications. The potential of CoT is not yet realised due to challenges in sharing and reusing IoT physical resources across multiple applications. Existing approaches provide small-scale and hardware-dependent shared access to IoT resources. This paper considers using market mechanisms to commoditise CoT resources as the approach to enable shared access to CoT resources and to improve their reusability. In order to achieve this, the requirements for trading CoT resources are discussed to conceptualise the proposed approach. A generic description model for CoT resource is introduced to quantify the value of CoT resources. In this paper, a marketplace architecture for trading CoT resources referred to as AMACoT is proposed. By formulating the trading of CoT resources as an optimisation problem, the proposed approach is experimentally validated. The evaluation measures the system performance and verifies the optimisation problem using three evolutionary algorithms. The evaluation of the optimisation algorithms demonstrates the optimality of trading CoT resources solutions in terms of resource cost, resource utilisation, provider lock-in and provider profit.**

Index Terms—**Cloud of Things, IoT, Marketplace, Optimisation**

I. INTRODUCTION

Cloud of Things (CoT) integrates Cloud Computing and Internet of Things (IoT) to complement each other. It aims to address the requirements of emerging IoT applications where either technology alone does not work. This includes applications that require Cloud resources but also interact with real-world objects/events that are far away from Cloud data-centres. The adoption of CoT is hindered by many challenges despite its benefits. These include latency, security [2] and resource management of IoT physical resources [3], [4].

Challenges in CoT resource management include load balancing, resource allocation and scheduling [5]. The complexity of resource allocation in CoT comes from the following. The heterogeneity of IoT resources, the constrained capacities of IoT resources, the scalability of IoT systems and the dynamism of IoT resources. These challenges lead to missing/limiting shared access to the IoT physical resources and consequently lessen the reusability of the resources across multiple applications. These also result in over-provisioned IoT infrastructure

A. S. Alrawahi is with the School of Science and Technology, Nottingham Trent University, Nottingham, UK, NG11 8NS and with Nizwa College of Applied Sciences, Nizwa, Oman, P. O. Box 699, PC: 611. E-mail: ahmed@alrawahi.org

A. Lotfi is with the School of Science and Technology, Nottingham Trent University, Nottingham, UK, NG11 8NS. E-mail: ahmad.lotfi@ntu.ac.uk

K. Lee is with School of Information Technology, Deakin University, Melbourne, Australia, VIC 3125. E-mail: kevin.lee@deakin.edu.au

to meet the unpredictable application requirements in which resources may be significantly underutilised.

This paper aims to improve shared access to CoT resources by proposing a marketplace architecture which can provide efficient resource allocation and deal with the complex issues present in the CoT. The solution re-imagines CoT resources as commodities rather than as organisational assets. It considers the business model of a marketplace whereby consumers request access (lease) to providers' resources. A marketplace that potentially can automate the trading between CoT resources and CoT applications. The proposed approach is motivated as follows. The deployments of IoT remain costly despite the increasing demand for IoT applications, the reduction in software and hardware costs [6]. This makes IoT adoption infeasible to many prospective users and emerging applications. Commoditisation of CoT enables the development of IoT applications independently from the infrastructure deployments. It also enables shared access to IoT physical resources. These likely lead to cost reduction in IoT resources and management as well as empowering new IoT applications and services.

There are various use cases for the CoT marketplace. For instance, an event management agency manages event facilities in a metropolitan area where it aims to improve its operational efficiency. The agency wants to develop an application that performs the following tasks. Find the least congested routes to an exhibition centre leavers. Crowd monitoring of fans attending games in a nearby stadium for better incidents response. Waste monitoring to efficiently automate the waste collection after organised events if needed. Having a dedicated CoT infrastructure for this application may require a significant upfront investment. In CoT marketplace, the agency would request a bundle of CoT resources to perform the tasks. For instance, footfall sensors and motion detection cameras around event facilities to guide people to the least congested tracks. The bundle may include sensors, actuators, cameras and other resources. The application consumes the required resources for a specific time and then releases them back to the marketplace when lease-time elapses. In this case, the application utilises the required resources without considerable investment nor dedicated infrastructure. Similarly, providers deploy their CoT resources without being tied-up to particular applications.

The remainder of this paper is organised as follows. The related work is reviewed in Section II. Section III describes the trading requirements of CoT resources followed by the description of the proposed marketplace architecture in Section IV. Section V analyses the security threats to AMACoT. The experimental setup and evaluation are presented in Section VI.

Section VII concludes this paper and describes the planned future work.

II. RELATED WORK

Market-based mechanisms for trading Cloud and IoT resources are intensively studied. Cloud-based approaches are more mature than the ones dedicated to IoT that are still emerging. As CoT relies heavily on IoT resources, this section focuses on IoT related market-based mechanisms. It reviews the recent literature and categorises them into one of the two categories described in the following sections.

A. IoT Marketplace Architectures

The concept of a marketplace for IoT-related resource is gaining prominence. A marketplace architecture for trading IoT data in real-time is proposed in [7]. The architecture enables providers to offer their IoT data streams for consumption by IoT applications. The proposed work differs from others by implementing the architecture and addressing various aspects, including scalability and compatibility. A generic Cloud-based marketplace architecture is proposed in [8] to enable trading of IoT deployments. The architecture addresses the Cloud-IoT integration and vendor lock-in issues. A marketplace for IoT resources is introduced as part of the broader architecture of the IoT ecosystem in [9]. The trading model of IoT information and functions is presented as a solution to five IoT interoperability issues.

A different approach is taken by [10] to establish a decentralised marketplace for IoT data based on blockchain technology. The proposed marketplace architecture uses simple contracts to simplify the trading of IoT data among participants. IDMoB is another decentralised marketplace for IoT data built on blockchain [11]. The marketplace enables trading of IoT data for none time-sensitive IoT applications. DataBroker DAO is another blockchain-based marketplace implemented to trade IoT data [12]. IoT generated data is traded using smart contracts among buyers and sellers via the Ethereum network.

A decentralised peer-to-peer marketplace for IoT data is presented in [13]. The proposed architecture differs from others by adopting fog computing model and blockchain technology. IoT data is prepared (e.g. filtered, processed) at a fog nodes layer while traded directly among the marketplace participants at the application layer. Another decentralised architecture is designed in [14] with the focus of IoT traffic metering and contract compliance. The presented system aims to improve transparency, fairness and interoperability while reducing the cost. The study also conceptualises a tracking model for the traded IoT data flows between IoT and Cloud Computing.

B. IoT Trading Mechanisms

Several market-based approaches are depicted and discussed in [15]. The study also proposes a game-theory based model to study the pricing of two IoT sensing services. Another set of market-based mechanisms is studied in [16]. This includes an analysis of IoT marketplace incentives, service patterns, information timeliness and social impacts. Two bidding algorithms to support IoT resource trading are introduced in [17]. The first algorithm aims to maximise the provider's revenue while the second is to lock the highest bid in the recurrent auction. Both algorithms are intended to protect the marketplace from collapse in certain trading scenarios.

A semantic matching model for IoT marketplace is presented in [18]. The model facilitates the matching process between providers' offerings and consumers' queries in a marketplace of the BIG IoT project [19]. A composition mechanism for IoT offerings is presented in [20]. The approach is based on a web-semantic model to describe IoT things and services for trading.

A feedback mechanism is proposed in [21] to support IoT data marketplaces. The blockchain-based approach enables consumers to rate the providers who have to maintain and improve their reputation based on feedback received. A marketplace model is also proposed in [22] to support the quality of trading. It introduces a credibility rating mechanism for providers based on the quality of their data. Another IoT marketplace model based on Stackelberg game is presented in [23] to model the trading processes in IoT environments. The model aims to minimise the complexities for IoT consumers while trading with IoT providers.

To support automation in IoT trading, a blockchain-based automated payment system is proposed in [24]. The system uses Ethereum contracts to automate payments without a need to intermediaries. An architecture is presented in [25] to support service level agreements (SLAs) while trading IoT data. The architecture uses three criteria model to improve satisfaction, payment and the SLA. Security mechanisms for protecting the IoT data marketplace is studied in [26]. The study also proposes authentication and authorisation model to control access to the traded resources.

C. Gap Analysis

The related work presented above shows it is merely focused on commoditising IoT data-sets/streams but not on IoT physical devices. Further limitations of the existing work are surveyed in [27]. This includes limited IoT resources integration, resource sharing and interoperability. Further limitations of existing shared access to IoT resources revolve around small-scale and hardware-specific support mechanisms [28], [6]. Our approach fills that gap by proposing a generic description model for CoT resources and a scalable architecture for trading them. To the best of authors' knowledge, these are the first dedicated to trading CoT resources.

III. TRADING CoT RESOURCES

Trading CoT resources is a multifaceted process. It involves mapping resources to applications, optimising the proposed maps, performing resource allocation and scheduling of the optimal map. To shape this approach, the following considerations are taken into account.

1) CoT resources and applications are heterogeneous. To decompose this complexity and improve interoperability, resource deployment and application development are considered distinct from each other. This lowers the
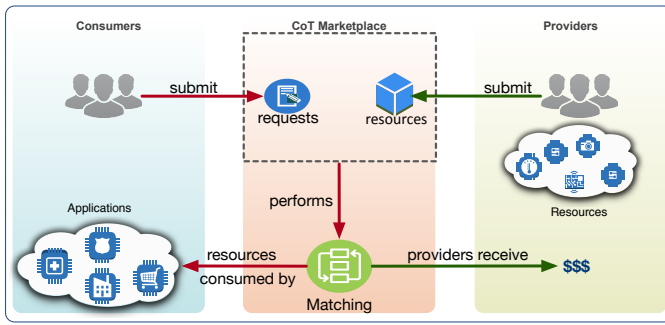
Figure 1. CoT marketplace concept.

investment required for infrastructure deployments, accelerate application prototyping and enable efficient trading.

2) CoT systems are large-scale and their commoditisation mechanisms, therefore, have to be scalable. In particular, having the ability to handle significant large numbers of resources and application requests simultaneously.

3) CoT resources are naturally constrained in terms of computing and power capabilities. Due to such challenge, concurrent shared access to those resources may not be possible but has to be resolved.

Based on the above-mentioned considerations, the following requirements of the proposed approach are identified.

A. The marketplace system

CoT marketplace concept is illustrated in Figure 1. The idea is to have a marketplace - denoted by M - where providers offer their deployed CoT resources and applications request access to the offered resources. The marketplace matches requests to resources by forming a bundle of resources from multiple providers based on the application requirements. More details about the marketplace architecture are provided in the proceeding Section IV. Table II provides trading vocabularies used with their description.

B. The marketplace participants

The marketplace consists of two categories of participants namely providers and consumers. Providers are CoT infrastructure owners/deployers and are denoted by $P = (p_1, \ldots, p_m)$. p_m represents an individual, organisation or a broker who manages resources on behalf of others. P submit their resources $R = (r_1, \ldots, r_j)$ to M.

Consumers are application owners/developers and are set to $C = (c_1, \ldots, c_n)$. c_n represents an individual, organisation or a broker who manages applications on behalf of others. C submit requests $RQ = (rq_1, \ldots, rq_i)$ to M for their CoT applications $A = (a_1, \ldots, a_z)$ to access and utilise a set of R.

C. A multi-attribute description model for CoT resource

The heterogeneity of CoT resources poses a challenge in defining and describing what a resource is? How can a wide range of heterogeneous resources be described generically? How can the value of CoT resources be quantified to enable

monetising them? The solution to these challenges is to propose a description model to 1) define CoT resource, 2) provide a generic description of their properties and 3) quantify the resources based on their described properties. The model is described as follows.

CoT resources can be defined generically based on their main physical components and functionalities. A CoT resource/node/device/thing is defined as a device with one or more basic computing functionalities (I/O, Processing, Storage) that interacts with its surrounding (using Sensor(s), Camera(s), Actuator(s)) and communicates with other entities of a network (using Communication unit(s)).

The model uses a multi-attribute description of the resources based on their physical and functional properties. Each resource r_j has a set of attributes $RA = (ra_1, \ldots, ra_t)$ that contributes to its monetary value when traded as a commodity. This includes multi-attributes of physical components (e.g. processing, actuating, sensing, power) and non-physical functions or features (e.g. security, location, redundancy). Each property or feature can be expanded into a multilevel sub-attributes to improve the presence of the commodity resource in the marketplace. A snapshot of a single resource description is provided in Table I.

The value of the multi-attributes presented in the previous step can be quantified by assigning corresponding numerical values. This is a vital step to monetise heterogeneous resources generically and to enable them to be traded. This can be achieved by assigning lower numerical values to attributes representing low resource specification and vice versa. Zero is a corresponding value for a missing resource component while a positive number corresponds to an attribute. This can be formulated as follows.

$$ra = \begin{cases} > 0, & \text{if } ra_j \text{ can be described from } r_j \\ 0, & \text{otherwise} \end{cases} \qquad (1)$$

For instance, a power capacity attribute of r_j can be quantified as either $[Power = [Permanent, 2]]$ or $[Power =$

Table I
SNAPSHOT OF CoT RESOURCE DESCRIPTION.

Components	Attributes	
	Properties	Example Values
Processor	Clock speed	320MHz
Memory	RAM	256KB
	Flash	1MB
Sensor(s)	Type(s)	Footfall, light
	Sensing Range	20 meter
	Max. Transmission Power	13.5dBm
	Num. of sensors	4
	accuracy	+/- 0.2 meter
Actuator(s)	Type(s)	Light
	Num. of Actuator	1
Camera	Type(s)	Motion detection
	Num. of cameras	1
Communication	Type(s)	WiFi
	Protocols	IEEE 802.11b/g/n
	Bandwidth	16Mbps
Power	Mode	Permanent
	capacity	5V
Security	Not available	0

$[Battery, 1]]$. In this case, a resource with permanent power supply is assigned a higher value than the one operates by a battery and is likely to have a better monetary value when commoditised. This offers the flexibility required in quantifying the value of heterogeneous CoT resources before the trading process starts.

D. The optimisation model

After CoT resources being described and their value quantified, resources can be matched with the application requests. This paper takes a unique approach to perform the matching by using gradient-free optimisation algorithms as a mapper for resources/requests. Using this type of optimisation algorithms (a.k.a. derivative-free algorithms) is justified as follows. This type of optimisation does not require the calculation of the gradient or derivative to find the optimal solution. The computation of the gradient can be impractical or computationally costly for large-scale optimisation problems such as trading CoT resources. This approach provides the following advantages that justify the use of optimisation algorithms.

1) Using optimisation algorithms provides improved architectural flexibility over a specially designed component (e.g. auctioneer, mapper). This means there is minimal or no need to change the optimisation algorithm to support any changes in other marketplace system components or CoT resources/requests.
2) Optimisation algorithms are known to find optimal solutions to NP-hard problems similar in complexity and scalability to the trading CoT resources problem presented in this paper.
3) The speed of many optimisation algorithms meets well with the requirements of the CoT marketplace in trading CoT resources promptly.

The optimisation model is presented in Figure 2 and described as follows. The optimiser receives resources from the providers and requests from the consuming applications. Both resources and requests are filtered to generate a potential map of matched resources to requests. Once the map is formed, the optimiser evaluates the map by minimising or maximising one or more trading objectives to find the most optimal set of requests matched to resources. The evaluation process continues till either the maximum number of iterations reached or specific criteria met (e.g. time elapsed, no improvement in finding optimal solutions for many iterations). Further description of the optimisation model is presented in Section IV.

E. Trading objectives

Trading objectives represent the goals of providers P and consumers C from participating in trading CoT resources. These goals are formulated as objective functions to provide significant flexibility for the trading model. Using this approach minimises the re-development effort of the system components that may be required in case of resource changes. Changes can be implemented as a new objective function without or with minimal changes in the system side.
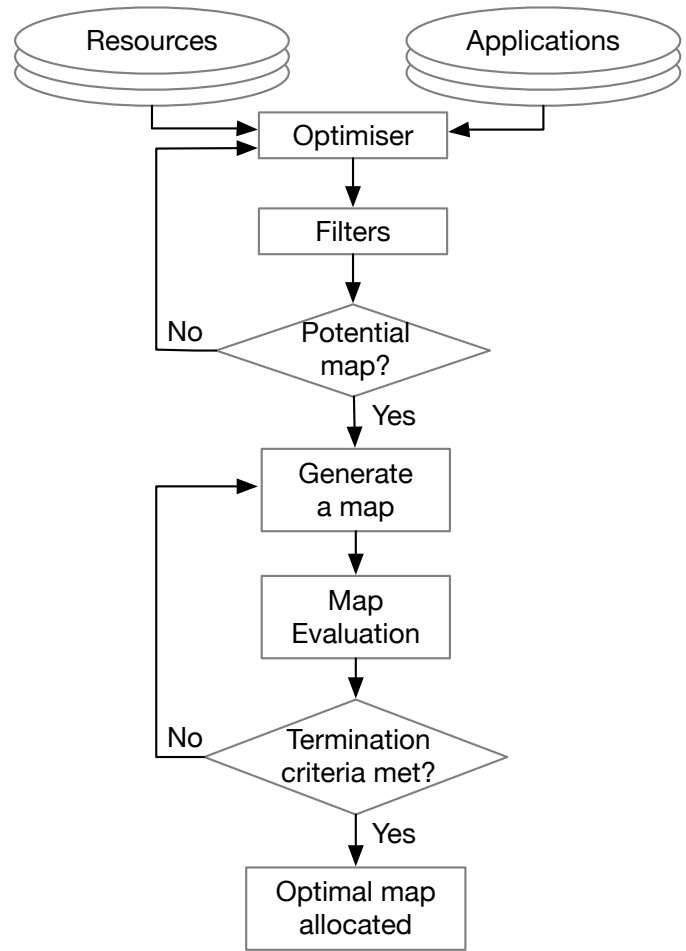

Figure 2. The optimisation model.

Table II
TRADING VOCABULARIES USED

Vocabulary	Description
M	the marketplace
P	providers $(p_1, ..., p_m)$ in M
R	traded resources $(r_1, ..., r_j)$ in M
C	consumers $(c_1, ..., c_n)$ in M
RQ	requests $(rq_1, ..., rq_i)$ submitted by C
A	applications $(a_1, ..., a_z)$ of C to utilise R
RA	resource attributes quantified numerically
cs_j	cost of a resource
b_i	a bid from consumer
t_i	utilisation time of resource requested by consumer
tq_{ij}	estimated transmission and delay time b/w a resource and application
rp_j	reputation of a provider
pu_i	requested utilisation time of a resource
ac_j	available resource components
uc_i	utilised components of a resource
py_j	provider policy
pt_j	proprietary technologies of a provider
mc_j	the marketplace charges

The trading of CoT resources is presented as a multi-objective optimisation problem as follows:

$$\text{Minimise} \quad CS = \sum_{i=1}^{n} \sum_{j=1}^{m} rc_j \cdot (t_i + tq_{ij}) \cdot rp_j \quad (2)$$

$$\text{Maximise} \quad RU = \sum_{i=1}^{n} \sum_{j=1}^{m} pu_i \cdot (ac_j - uc_i) \quad (3)$$

$$\text{Minimise} \quad P_{lock} = \sum_{j=1}^{m} py_j + pt_j \quad (4)$$

$$\text{Maximise} \quad PR = \sum_{i=1}^{n} \sum_{j=1}^{m} rc_j \cdot (t_i + tq_{ij}) - mc_j \quad (5)$$

$$\text{subject to} \quad 0 < cs_j \leq b_i \quad (6)$$
$$0 < Er_i \leq Ep_j \quad (7)$$
$$se_i \geq se_j \quad (8)$$
$$rp_i \geq rp_j \quad (9)$$
$$ra_i \geq ra_j \quad (10)$$

where $i = 1, ..., n; j = 1, ..., m$ for Constraints 6, 7, 8, 9 and 10. Descriptions of the objectives and constraints are provided below:

Objective 1. Minimising the resource cost is one of the usual motivations of the consumers. Consumers are likely to bid for minimal cost resources. The cost objective function is presented in Function 2. The following contributors to the total cost CS are considered when minimising the cost of requested resources. Let b_i be the bid from a consumer and cs_j the provider's cost. The initial cost rc_j can be calculated as $rc_j = (b_i - cs_j)$. The requested utilisation time of a resource is set to t_i while tq_{ij} denotes the estimated transmission and delay time. Provider reputation rp_j is set based on the credibility measures of the marketplace to determine the trustworthiness of the provider. The reputation is assumed as part of the resource cost for two reasons. 1) It enables the marketplace to use any feedback mechanism that allows consumers to rate their providers' trustworthiness. 2) The reputation of the provider has an indirect effect on the cost of resources. A provider with a higher reputation is enabled to offer its resources with better cost than a low-rated provider.

Objective 2. CoT applications are assumed to monopolise access to a set of resources for a given time[29]. This can cause low utilisation of the allocated resources due to the light requirements of some applications. For trading CoT resources to be efficient, resource utilisation has to be optimised. Resource utilisation objective function is presented in Equation 3. The objective considers the requested resource utilisation time pu_i, the available resource components ac_j and the actually utilised components of a resource uc_i.

Objective 3. Function 4 presents the objective of minimising provider lock-in. Vendor lock-in is a common challenge for commoditised computing services. It describes the situation where consumers can not migrate their data or applications to different providers due to various reasons. This objective aims to minimise the lock-in by considering the provider policy py_j that enables consumers to migrate and the proprietary technologies of the provider pt_j. Both factors are rated from

$1 - 5$, where 1 is the most flexible policy towards consumer migration and lowest proprietary technologies that may hinder consumers from migrating to different providers.

Objective 4. Providers always aim to maximise their profit PR. Equation 5 presents the profit objective function. This can be achieved by maximising the cost of resources rc_j and their utilisation time t_i while considering the marketplace charges mc_j as expenses.

In addition to the objectives, constraints are used to identify feasible solutions to the resource trading problem. This significantly minimises the search space of such scalable and complex set of candidate solutions. Constraint 6 illustrates that costs and bids have to be positive, and bids are always greater than or equal resource costs. The energy constraint presented in Constraint 7 ensures the required energy Er_i to perform application tasks does not exceed the available resource energy Ep_j. Constraint 8 specifies the security requirements of the application se_i to be satisfied by the security capabilities of the resource se_j. Constraint 9 provides credibility insurance to the marketplace participants based on their performance. Providers have to maintain a certain reputation level rp_j in the marketplace while consumers specify their providers' credibility requirements rp_i. Constraint 10 enable participants to specify participant-specific requirements or limitations in responding to some applications or resources attributes. This constraint is part of this approach genuineness and flexibility to handle heterogeneous CoT resources and applications.

F. Other requirements

Other requirements of the marketplace can be implemented based on its design and operational goals. This paper justifies those requirements as follows.

Scalability. A marketplace for CoT resources is expected to handle a considerable number of consumer requests and provider resources simultaneously. This can impact system performance and consequently fail to trade CoT resources. The marketplace system should have the ability to handle various scales of consumer requests and provider resources.

Security. Security is vital for any marketplace architecture. Security analysis is performed in Section V to address the potential security threats to the proposed system.

Multiple Business Models. To satisfy the requirements of different CoT applications, the marketplace should support different business models. This includes consumer-to-consumer, provider-to-consumer and provider-to-provider. It may also consider implementing the system to support one or more of the following market structures: broker system, monopoly market, oligopoly market, single-side auction and double-side auction.

Multiple Participant Objectives. Marketplace participants have different requirements/goals. For instance, consumers are likely to bid for the lowest cost possible while providers aim to maximise their revenues. Consumers may have conflicting objectives at the same time (e.g. maximising resource coverage while minimising the response time). The marketplace system should provide a mechanism to maintain the balance among conflict objectives.
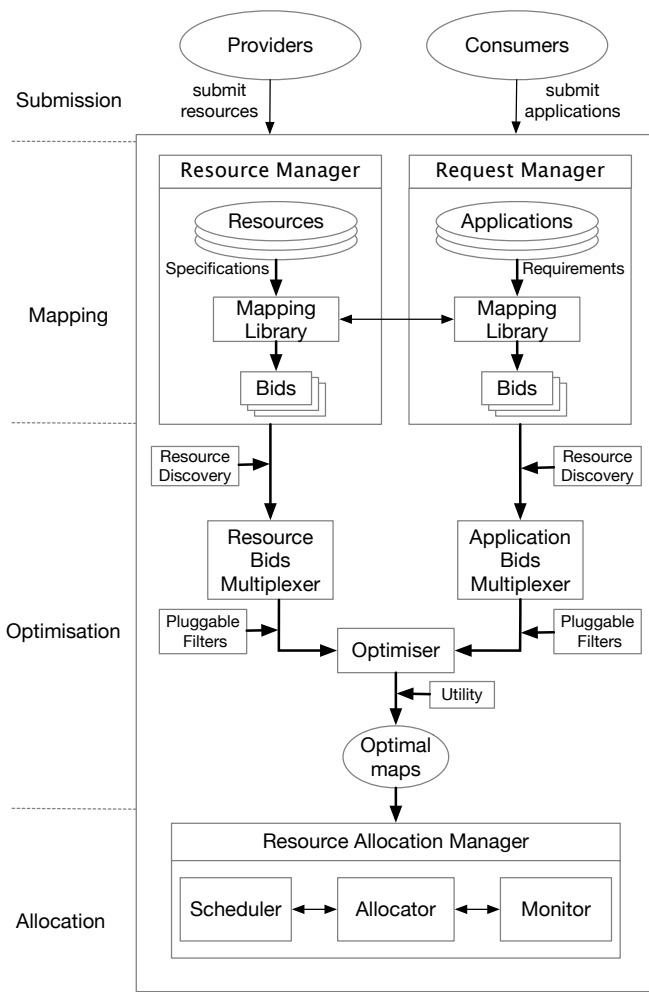
Figure 3. CoT Marketplace Architecture (AMACoT)

IV. THE MARKETPLACE ARCHITECTURE

The proposed marketplace architecture is illustrated in Figure 3. The architecture is structured into four functional layers as follows. Submission layer represents the marketplace entry point where participants are authenticated and granted authorised access to trade.

The mapping layer consists of resource and request managers. Resource manager provides interfaces that enable resource providers to submit, update and remove their resource specifications. Resources are described and quantified based on the description model discussed in Section III-C. Resource descriptions include connectivity options and resources are assumed to be connected already to the Internet via IoT gateways. Similarly, the request manager's interfaces receive application requirements from consumers. Application requirements are high-level descriptions of the computing and budget needs as illustrated in Table III. Consumers can also update and remove their applications using the request manager. Mapping libraries of both resource and request managers provide early local coordination to turn resource specifications and application requirements into bids.

Optimisation layer represents the operational tier of the

system. Applications can discover resources that are already stored in the resource manager. Bids flow from resource and request managers to multiplexers where they are selected to form bundles. Resource bundles represent a set of resources from multiple providers that can potentially be utilised by multiple applications. Whilst being forwarded to the optimiser, resource bundles and application requests are filtered. Pluggable filters include a wide range of filtering criteria such as location, resource coverage, computing and energy requirements. Filtered resource bundles and application requests are forwarded to the optimiser. The optimiser performs a two-stage process as follows; 1) construct optimal maps that consist of resource bundles and application requests ready for allocation, 2) evaluate the optimal maps based on the participants' goals using utility functions presented in Section III-E. One optimal map is forwarded to the resource allocation manager for resources to be allocated to the applications.

Allocation layer consists mainly of the resource allocation manager. The scheduler manages the utilisation time of the resources based on the application requirements. It also coordinates with the allocator to enable resources joining the application network and dis-joining when the lease-time elapses. The monitor captures resource allocation events in real-time and provides interfaces where consumers and providers oversee their transactions.

V. THREAT ANALYSIS

The marketplace system should enforce different security measures to secure its operations. Security threats are analysed using STRIDE model [1] to help the design of the architecture by identifying potential threats. STRIDE model is used due to its maturity among other threat modelling techniques and due to its simplicity. Table IV illustrates the STRIDE threats, security propriety violated and the impacted layers of the proposed architecture.

As Table IV shows, each layer of AMACoT components may be impacted by one or more type of threats. At submission layer, an attacker can illegally gain access and use a consumer's or a provider's credentials to access the marketplace. AMACoT can mitigate this threat by using authentication protocols that do not require a password or use signed certificates to verify the authenticity of consumers/providers. The attacker can also tamper at the submission layer by maliciously modify a consumer's request or a provider's offerings. These types

Table III
SNAPSHOT OF CoT APPLICATION REQUIREMENTS.

Requirements	Example Value
Processing	$\geq$ 1GHz
Memory	$\geq$ 1GB
Storage	Any
Network	Heterogeneous
Energy	Battery/Permanent
Sensing	Environmental
Actuator	0
Security	$\geq$ Basic
Location	[52.95610793607633, -1.1453494058431906]
Provider's Rating	$\geq$ 3/5
Budget	$\leq$ $10 per hour

of threats may occur using bit-flipping or injection attacks. AMACoT can mitigate these attacks by integrating adequate users' input/output validation tools for proper data integrity validation. Submission layer is also susceptible to denial of service (DoS) attacks where the attacker aims to interrupt the marketplace making it unavailable or unstable to providers/consumers. This can occur when the system is flooded with a large number of concurrent requests. The security manager can alleviate DoS attacks by employing requests/offers limiter to maintain the number of submissions at an acceptable level.

The mapping layer can be vulnerable to the threats of information disclosure and elevation of privileges. Information disclosure threats the confidentiality of marketplace users when the attacker maliciously gets hold of the users' sensitive data stored in the resource manager and/or the request manager. AMACoT can use a common practice to mitigate this threat by encrypting users' sensitive data. Elevation of privileges also poses a considerable risk at the mapping layer. An attacker can attempt to gain some privileges that enable him to perform some actions that he cannot achieve. This may include manipulating bids at either resource or requests manager or both. The system should implement robust authorisation techniques and operate the components at the mapping layer using non-root users. The optimisation layer can also be vulnerable to both threats, in addition to the DoS attacks. AMACoT can prevent such attacks using the same mitigating mechanisms discussed earlier for the submission and mapping layers.

The resource allocation manager is vulnerable to data tampering. This may occur when a user tries to manipulate a schedule before resources are allocated to take advantage of other users. Mitigation may include validating users' input/output to detect and prevent data tampering.

Although the security aspects are crucial to the marketplace architecture, the focus of this paper is on demonstrating the feasibility and performance of AMACoT in CoT resource allocation using optimisation algorithms. For deployment of this architecture if it is important to take security as the aim of the study. It would be necessary to take standard IoT security precautions such as those identified in [36] Those precautions and any deployment of specific security mechanisms, therefore, fall out of this paper's scope.

VI. EXPERIMENTAL EVALUATION

This section presents the experimental evaluation of the proposed work. After the experimental setup explanation in Section VI-A, Section VI-B provides system verification aiming to evaluate the system footprint and Section VI-C presents algorithmic evaluation of proposed approach using different optimisation algorithms.

A. Experimental Setup

The architecture is developed using Python in a computer with 2.3 GHz Xeon processor, a 125GB memory and Linux OS. In order to simulate the behaviour of distributed systems, Python Remote Object (Pyro4) is used to connect the main components of the system as well as consumers and providers.

The optimisation engine of AMACoT integrates optimisation algorithms to map requests to resources and to evaluate the optimal resource allocation. The following optimisation algorithms are implemented as follows.

NSGAII. Non-dominated Sorting Genetic Algorithm II [30] is an improved genetic algorithm that is widely used in real-world multi-objective optimisation applications. The population size is set to 200 with a maximum number of 200 iterations for all experiments.

NSGAIII. This algorithm is an extension of the NSGAII that uses reference points to diversify the Pareto points during the search[31]. Besides the same settings used for NSGAII, the number of divisions is set to 12.

SPEA2. Strength Pareto Evolutionary Algorithm 2 [32] is designed and used to optimise combinatorial problems. The population size is set to 200 with a maximum number of 200 iterations for all experiments.

B. Implementation Verification

Stress tests are performed to evaluate the footprint of the system components when they interact with each other as well as interacting with providers and consumers. Three experiments are performed using three different scale factors as shown in Table V. The scale factors aim to evaluate the scalability of AMACoT system and measure the overheads generated. In these three experiments, SPEA2 algorithm is used to minimise the resource cost while maximising the provider profit. This evaluation measures the following system footprints; 1) CPU usage, 2) Memory usage, 3) Latency that is measured from the time of request submission to the time of resource allocation confirmation and 4) Throughput to measure the number of requests/resources handled by AMACoT over the trading time.

Experiment 1. $10,000$ requests and $200,000$ resources are submitted to AMACoT by 100 consumers and 100 providers respectively. Experiment 1 requires 10% of CPU, 3GB of memory and 57 second to produce an optimal resource allocation. AMACoT handles 175 requests and over 3500 resources per second.

Experiment 2. $20,000$ requests and $400,000$ resources are submitted to AMACoT by 200 consumers and 200 providers respectively. Experiment 2 consumes 11% of CPU, 7GB of memory and 119 second to produce an optimal resource allocation. The maximum throughput of this experiment is 168 requests and 3361 resources per second.

Experiment 3. $30,000$ requests and $600,000$ resources are submitted to AMACoT by 300 consumers and 300 providers respectively. The peak CPU load of Experiment 3 is 13% while 11GB of memory used. Producing an optimal resource allocation requires 185 second for experiment 3. AMACoT processed 162 requests and 3243 resources per second.

The results of the verification tests are summarised in Table VI. Results show that CPU usage increases from 10% in experiment 1 to 11% in experiment 2 when experiment 2 scales up by 100%. The CPU load also increases from 11% in experiment 2 to 13% in experiment 3 that scales up by 100%. This implies a reasonable CPU usage when

Table IV
STRIDE MODEL OF AMACoT

	Threat	Property Violated	Impacted Layer			
			Submission	Mapping	Optimisation	Allocation
S	Spoofing identity	Authentication	●			
T	Tampering with data	Integrity	●			●
R	Repudiation	Nonrepudiation				
I	Information disclosure	Confidentiality		●	●	
D	Denial of service	Avaiability	●		●	
E	Elevation of privilege	Authorisation		●	●	

marketplace participants increase significantly. Memory usage is also measured for the three experiments as follows. Experiment 1 requires $3GB$ of memory, $7GB$ for experiment 2 and $11GB$ for experiment 3. Memory consumption increases from $3GB$ in experiment 1 to $7GB$ in experiment 2 when the marketplace participants rise by 100%. In experiment 3, the memory consumption increases up to $11GB$ when the experiment scales up by further 100%. Results indicate a fair memory usage across the three experiments when different scale factors are considered.

Latency relies significantly on three aspects as follows. 1) The optimisation algorithm used, 2) The complexity of optimised objectives and 3) The number of optimised objectives. The latency results presented are obtained from SPEA2 algorithm optimising two objectives. The latency of experiment 2 is about 2 times the latency of experiment 1 while the latency of experiment 3 is about 1.5 times the latency of experiment 2. This implies that the latency is doubled as the experiment scales up by 100%. The throughput results show that request throughput decreases 4% only when the marketplace participants rise by 100% in experiment 2 from experiment 1. The request throughput declines 3.5% further in experiment 3 when compared to experiment 2. The resource throughput in experiment 2 shows 4% reduction in comparison to experiment 1 while it decreases 3.5% in experiment 3 when compared to experiment 2. The overall evaluation of throughput shows sensible throughput variations across the three experiments.

Table V
SIMULATED MARKETPLACE PARTICIPANTS.

Parameter	Experiment 1 (scale factor 1)	Experiment 2 (scale factor 2)	Experiment 3 (scale factor 3)
Number of Requests	10K	20K	30K
Number of Resources	200K	400K	600k
Number of Consumers	100	200	300
Number of Providers	100	200	300

Table VI
PERFORMANCE COMPARISON.

Parameter	Experiment 1	Experiment 2	Experiment 3
Peak CPU(%)	10	11	13
Peak memory(GB)	3	7	11
Latency(sec)	57	119	185
Throughput(Request/sec)	175	168	162
Throughput(Resource/sec)	3508	3361	3243

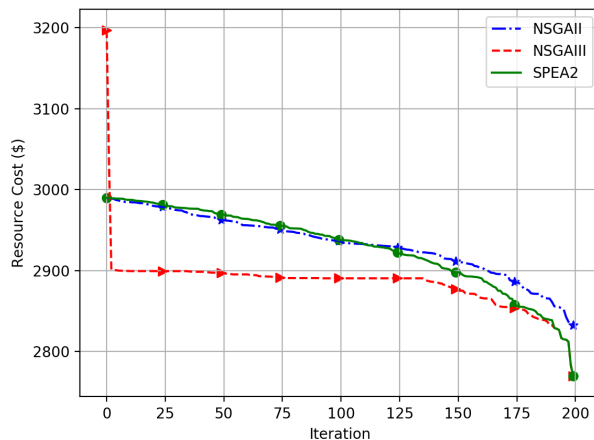Figure 4. Optimising the resource cost at the end of each iteration.

C. Algorithmic Evaluation

In order to evaluate the performance of the optimisation algorithms used and the quality of their optimal solutions, a set of ten experiments are performed using the same scale factor of experiment 3. The following experiments optimise a single objective and compare the results of the three algorithms used.

Experiment 4. This experiment minimises the resource cost as presented in Function 2. Figure 4 shows the comparative evaluation results for resource cost optimality. All algorithms compete towards optimal solutions but NSGAIII and SPEA2 find better cost than NSGAII.

Experiment 5. This aims to minimise the possibility of provider lock-in as presented in Function 4. Figure 5 illustrates that NSGAII and NSGAIII algorithms converged into an optimal solution that is approximately 24% lower than the solution of SPEA2.

Experiment 6. This experiment is intended to maximise the provider profit as described in Function 5. Figure 6 demonstrates the competition between NSGAII and SPEA2 in which both algorithms take approximately the same direction to the optimal profit. In contrast, NSGAIII improves its solutions significantly during early iterations and maintain steady improvements towards the last iteration. NSGAII and NSGAIII provide slightly better profit for providers than SPEA2.

Experiment 7. Resource optimisation is performed in this experiment to maximise the resource utilisation by consumers
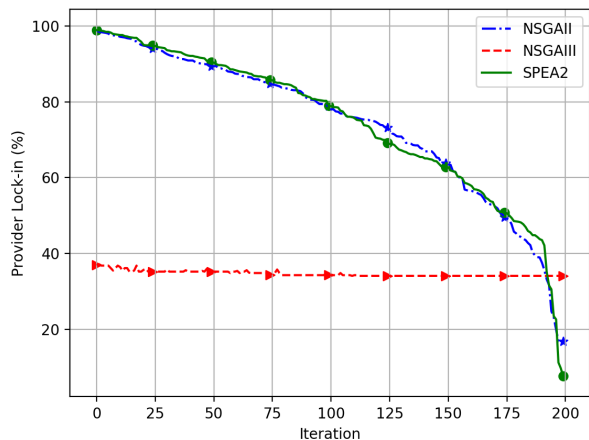
Figure 5. Optimising the provider lock-in utility at the end of each iteration. The provider lock-in rate is minimised, so consumers avoid being locked-in using resources from a single or very few providers.
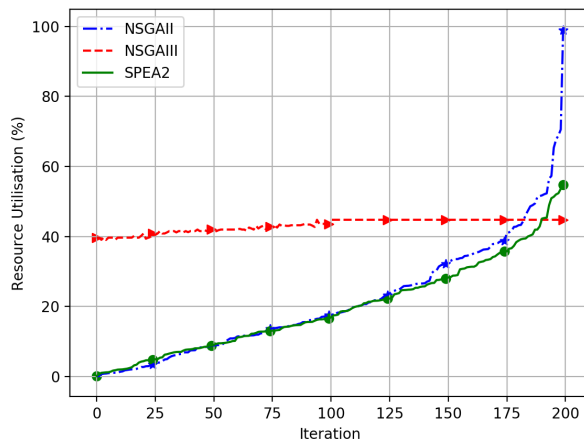

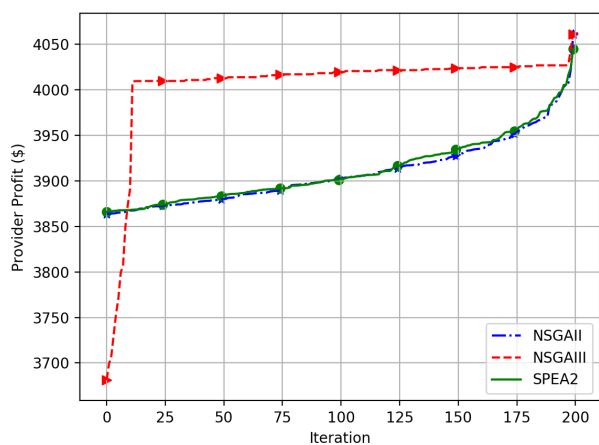Figure 7. Optimising the resource utilisation at the end of each iteration.

Figure 6. Optimising the provider profit at the end of each iteration.

Figure 8. Optimising the aggregated utility of resource cost and provider profit.

as presented in Function 3. Figure 7 illustrates NSGAII outperforms other algorithms in maximising the resource utilisation. In contrast to the other algorithms, NSGAIII shows insignificant changes throughout the process.

Experiments 4-7 perform the standard optimisation of a single objective that may not be practical for many real-world CoT applications. CoT applications often involve multiple objectives and therefore require multi-objective optimisation to find optimal solutions for two or more objectives including conflicting ones (e.g. min. resource cost while max. resource utilisation). This conflict is commonly addressed by using Pareto approach [33] to evaluate a set of trade-off solutions. The following experiments show the progression of optimising multiple objectives using different approaches.

Experiment 8. This experiment optimises resource cost and provider profit as presented in Functions 2 and 5, respectively.
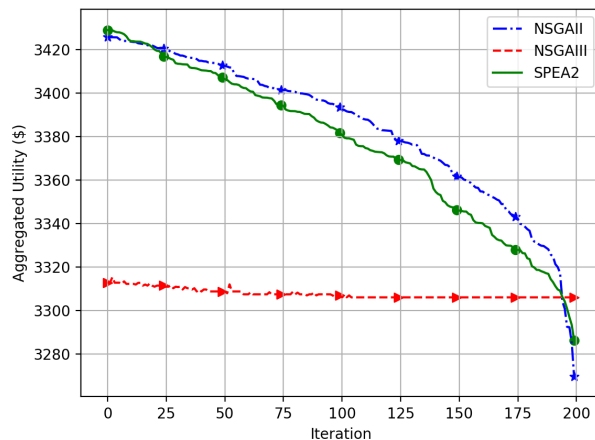
A way of performing this is to aggregate both objectives into a single one using weight factors as follows.

$$\underset{RQ,R}{\text{Minimise}} \quad AU = w_1 CS - w_2 PR \qquad (11)$$

$$\text{subject to} \quad 6, 7, 8, 9, 10 \qquad (12)$$

where w_1 and w_2 are the weights for resource cost and provider profit respectively. Weighting factors are used to prioritise objectives. To maintain the balance between the two objectives, the value of w_1 and w_2 is equally set to 0.5 where $w_1 + w_2 = 1$. Figure 8 shows that NSGAII and SPEA2 outperforms NSGAIII despite the lower start of NSGAIII.

Prioritising objectives for CoT applications is challenging for the following reasons. First, it requires prior knowledge of the problem to assign appropriate weights. This prior knowledge may not always be available to CoT applications. Second, optimisation objectives are application-specific and therefore
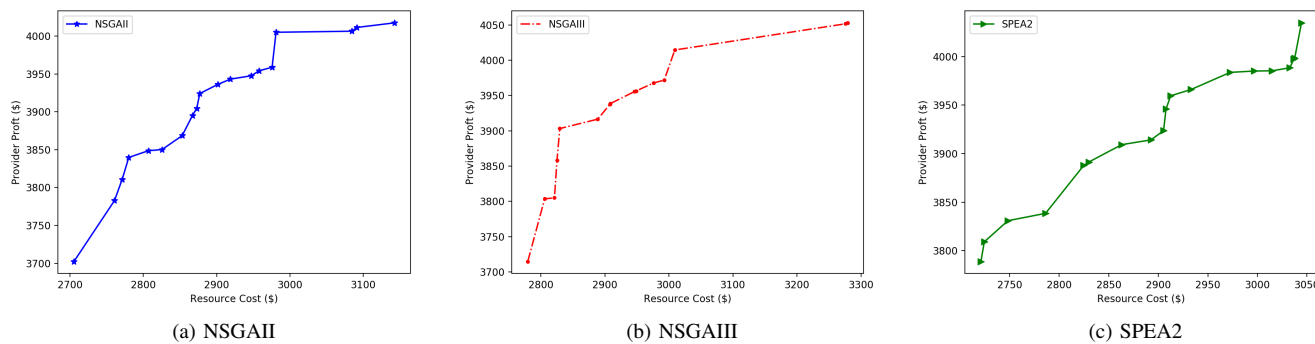
Figure 9. Optimising the resource cost and the provider profit. NSGAII produces the largest set of solutions, the lowest resource cost and the lowest provider profit. NSGAIII provides lowest set of solutions, the highest resource cost and the highest provider profit. SPEA2 yields various optimal solutions that maintain the balance when compared to the bi-objective optimality of the other algorithms.
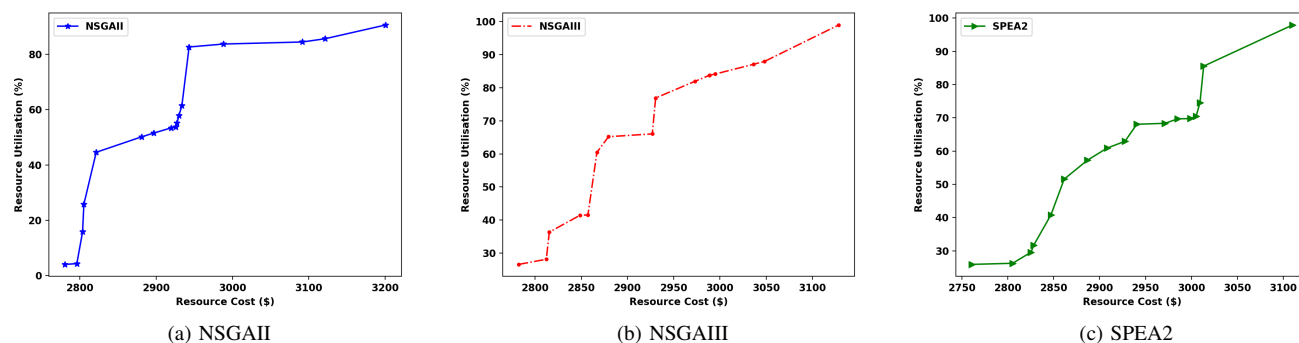

Figure 10. Optimising resource cost and the resource utilisation. NSGAII generates the lowest resource utilisation and the most expensive resource cost. NSGAIII produces the maximum resource utilisation while SPEA2 yields the minimum resource cost.

need re-prioritisation more frequently using weighting factors. Third, this approach yields one optimal solution only. This gives the decision-maker no other solutions to the problem. Although this method may benefit specific applications with prior knowledge about the problem, the following experiments consider using Pareto approach [33] to evaluate a set of optimal solutions rather than one solution only. Experiments 9-12 evaluate conflicting bi-objectives that reflect real-world business requirements. Using the Pareto approach produces a set of optimal solutions for both objectives where an optimal solution of an objective does not worsen the solution of the other objective. Using this approach aims to maintain the balance among conflicting objectives of consumers and providers.

The challenge is how to measure the quality of Pareto-generated solutions of different optimisation algorithms. To overcome this shortcoming, each set of optimal solutions produced in the following experiments is evaluated using the hypervolume indicator (HV) [34]. HV measures the size of the covered space by the generated set of Pareto solutions. A higher value of HV indicates the better distribution of the Pareto solutions and approximately closer to the optimality.

Experiment 9. This experiment optimises resource cost and provider profit as presented in Functions 2 and 5, respectively. Figure 9 shows there is an insignificant difference among

the optimal solutions of the three algorithms. The HV values of NSGAII, NSGAIII and SPEA2 are $0.57, 0.56$ and 0.62, respectively as shown in Figure 14. This implies that SPEA2 generates slightly better optimal solutions for minimising resource cost and maximising provider profit.

Experiment 10. Another business requirement for CoT applications is to optimise resource cost and resource utilisation benefiting the resource consumers. Experiment 10 is intended to minimise the resource cost and maximise the resource utilisation as described in Function 2 and 3, respectively. As illustrated in Figure 10, all algorithms provide multiple solutions that maintain the balance between minimising the cost and maximising the resource utilisation. It can also be noted that the resource cost increases as the resource utilisation increases. This implies there is a trad-off between resource cost and utilisation as resource providers may enable higher resource utilisation with higher cost. Figure 14 shows an overall high HV indicator for the three algorithms with slight differences among them.

Experiment 11. This experiment optimises provider profit and provider lock-in as presented in Functions 5 and 4, respectively. It aims to benefit both providers and consumers by maximising the provider profit while minimising the chance of consumers being locked in one or very few providers' infrastructures. Figure 11 shows that NSGAII and SPEA2
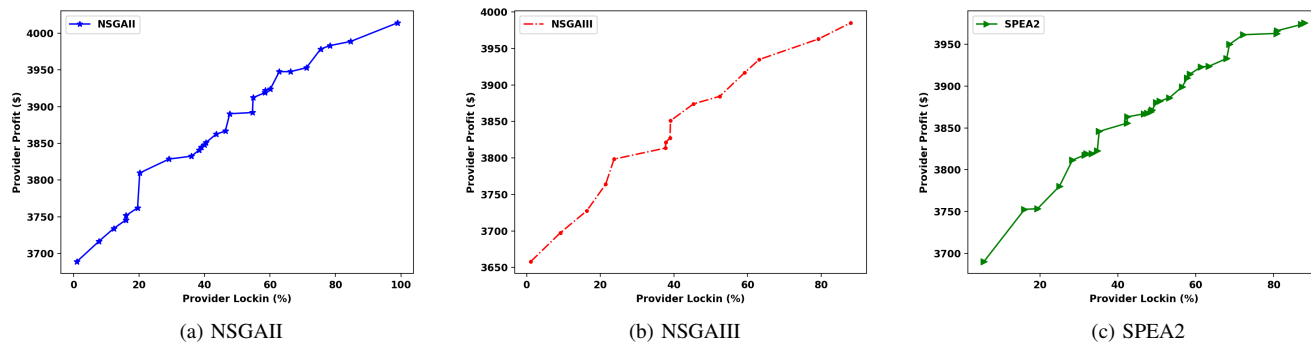
Figure 11. Optimising provider profit and provider lock-in. NSGAII provides the largest set of solutions, the the maximum provider profit and the highest provider lock-in. NSGAIII produces the smallest set of solutions and the minimum provider lock-in. SPEA2 maintans the balance between the number of generated solutions, the minimum provider lock-in and the maximum provider profit.
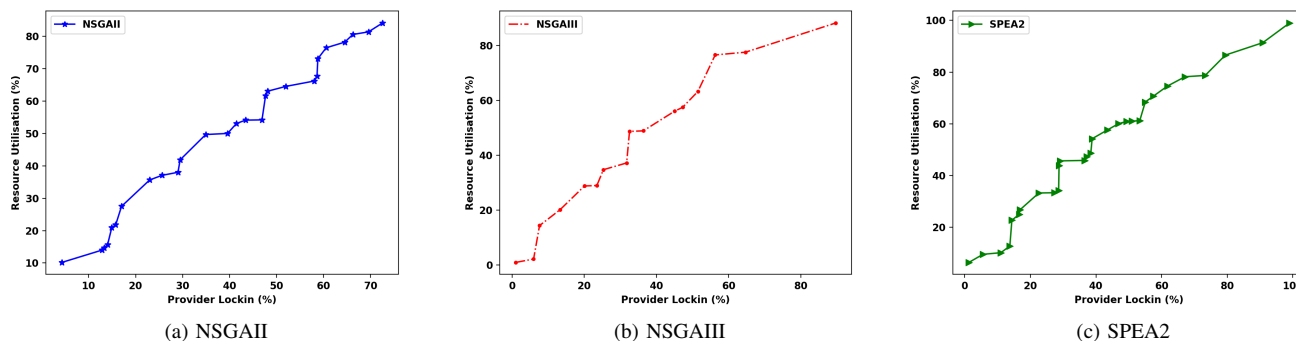

Figure 12. Optimising resource utilisation and provider lock-in. NSGAIII provides the smallest set of solutions, the lowest resource utilisation and the minimum provider lock-in. SPEA2 produces the largest set of solutions, the maximum resource utilisation and the highest rate of provider lock-in.

provide numbers of solutions that are approximately twice as NSGAIII does. NSGAIII, however, generates a similar distribution of Pareto-generated solutions. Figure 11a and 11c, respectively, illustrate that NSGAII and SPEA2 find over 50% of the solutions with provider lock-in rate of 40% or more. This may indicate providers' preference of locking consumers to maximise the profit. Figure 14 shows that NSGAII outperforms NSGAIII by 7% and SPEA2 by approximately 10%.

Experiment 12. This experiment addresses the requirement of CoT applications to optimise resource utilisation and provider lock-in as presented in Function 3 and 4. Figure 12 shows that all algorithms produce over 60% over their solutions with provider lock-in rate of 30% or more. This may imply that resources with high utilisation rates are associated with high chances of provider lock-in. Figure 14 illustrates that all algorithms attain similar HV values. This suggests a similar performance of the algorithms in finding the optimality of resource utilisation and provider lock-in utility.

Experiment 13. This experiment aims to optimise resource cost, resource utilisation, provider lock-in and provider profit and as described in Function 2, 3, 4 and 5. It explores the potential optimality of multiple conflicting objectives as well as the performance of the optimisation algorithms. Visualising the Pareto fronts of large-dimensional multi-objective optimisation problems is known to be a challenge [35]. One of the

ways to visualise the results of this experiment is to use the scatter plot matrix as illustrated in Figure 13.

In this experiment, Pareto fronts can be identified as shown in Figure 13d, 13e and 13f. Figures 13a, 13b and 13c show the solutions scattered across the solution space while Pareto fronts are not typically formed yet. This can be clearly seen in Figure 13b as a typical front should be formed towards the left side of both axes when both objectives are minimised. This may imply the following. 1) Generating Pareto optimal solutions is possible in the case of a high-dimensional optimisation problem. 2) The optimiser parameters may need to be improved to address the increase in the number of objectives. The number of iterations and the population size have not been changed in this experiment to be consistent with other experiments performed with the same parameters.

Experiments presented in this section demonstrate the progression of optimising various objectives. Experiments 4-7 optimise single objective, Experiment 8 optimises multiple objectives using the weighted sum method while Experiments 9-12 optimise multiple objectives using Pareto approach. Experiment 13 optimises all functions presented earlier using Pareto approach. Experiments 4-8 generate one optimal solution each while Experiments 9-13 provide a set of optimal solutions each. The evaluation of Experiments' results using HV indicator suggests the following. Experiment 8 (weighted
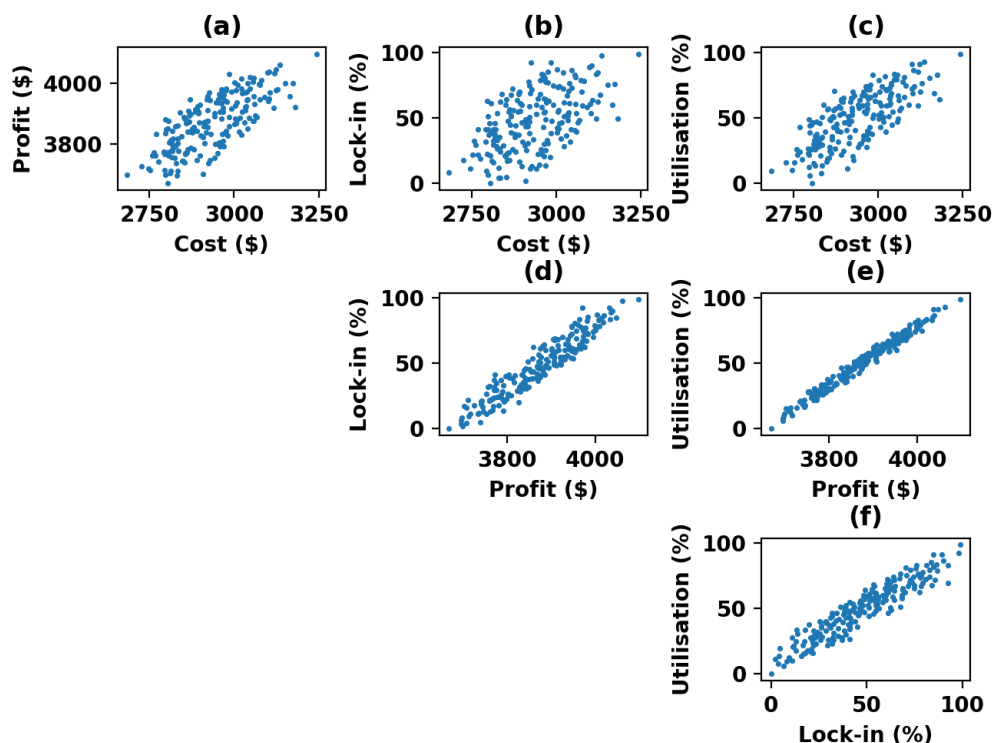
Figure 13. Scatter plot matrix showing Pareto solutions of all bi-objective combinations of Experiment 13.

AMACoT incurs insignificant CPU and memory overheads when marketplace participants are doubled in Experiment 2 and tripled in Experiment 3. AMACoT also maintains a good level of throughput with a minimal reduction below 5% across all the three experiments. The overall stress results imply the advantage of reducing the architectural complexity in CoT by using an optimisation algorithm as the core of the trading manager rather than a specific-purpose system component.

Latency becomes a limitation for AMACoT performance with respect to scalability. The latency increases by approximately two-fold when resources and requests are doubled. This may imply the dependency of optimisation algorithms on the hardware setup. Performing the same experiments in higher hardware specifications may reduce the latency of the system significantly. Running the optimiser on multi-processing setup can also provide further improvement to the system latency.

The evaluation of optimisation algorithms presented in Section VI-C provide comparative results of the algorithms performance and the optimality of their solutions. The evaluation validates the use of objective functions in quantifying the value of CoT resources. This implies the heterogeneity of CoT resources and the dynamic requirements of CoT consumers can be formulated as objective functions that are likely to be optimised. It can be concluded that using optimisation-based approaches as market mechanisms for CoT resource is feasible and promising.
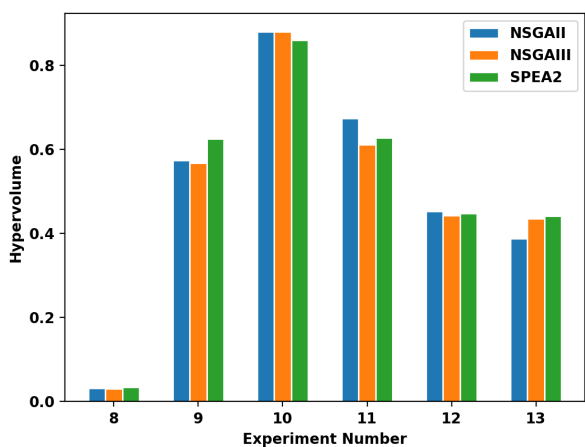

Figure 14. Evaluating the Pareto-generated solutions using HV indicator.

sum approach) produces the lowest HV score. This is likely because the approach produces one optimal solution only that cannot contribute to the volume calculation. In Experiments 9-13, HV values indicate that NSGAII performs better than the other algorithms on three experiments while SPEA2 outperforms others on two experiments.

D. Discussion

The performance results presented in Section VI-B show reasonable system overheads and demonstrate good scalability.

VII. CONCLUSIONS AND FUTURE WORK

This paper proposes a trading approach for CoT resources consisting of a multi-attribute description model for CoT

resources, the trading objectives and the marketplace architecture. In contrast to other approaches, the proposed approach separates between CoT application development and hardware deployment considering CoT resources as commodities. Experimental evaluation validates the system and algorithmic performance. AMACoT generates optimal solutions using single optimisation, weighted sum and Pareto fronts approaches. The optimality of resource cost, provider lock-in, resource utilisation and provider profit is evaluated. HV indicator is used to measure the performance of the optimisation algorithms and the quality of the optimal solutions they produced.

In order to address the limitations of this work, the following future work is planned. The adaptivity requirements of trading CoT resources will be investigated. Improving the system latency by potentially performing the optimisation on multiprocessor setup and exploring further trading objectives to address more consumer and provider requirements (e.g. QoS attributes).

REFERENCES

[1] Shawn Hernan, Scott Lambert, Tomasz Ostwald, and Adam Shostack. Threat modeling-uncover security design flaws using the stride approach. *MSDN Magazine-Louisville*, pages 68–75, 2006.

[2] C. Chang, S. N. Srirama, and R. Buyya, "Internet of things (iot) and new computing paradigms," *Fog Edge Computing: Principles Paradigms*, pp. 1–23, 2019.

[3] Z. Ning, X. Kong, F. Xia, W. Hou, and X. Wang, "Green and sustainable cloud of things: Enabling collaborative edge computing," *IEEE Commun. Mag.*, vol. 57, no. 1, pp. 72–78, 2019.

[4] M. Aazam, I. Khan, A. A. Alsaffar, and E.-N. Huh, "Cloud of things: Integrating internet of things and cloud computing and the issues involved," in *Appl. Sci. Technol. (IBCAST), 2014 11th International Bhurban Conference on*. IEEE, 2014, pp. 414–419.

[5] T. Kumrai, K. Ota, M. Dong, J. Kishigami, and D. K. Sung, "Multiobjective optimization in cloud brokering systems for connected internet of things," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 404–413, 2017.

[6] S. Akkermans, W. Daniels, G. Sankar R, B. Crispo, and D. Hughes, "Cerberos: a resource-secure os for sharing iot devices," in *Proc. of the 2017 International Conference on Embedded Wireless Syst. Netw.*. Junction Publishing, 2017, pp. 96–107.

[7] B. Krishnamachari, J. Power, S. H. Kim, and C. Shahabi, "I3: An iot marketplace for smart communities," in *Proc. of the 16th Annual International Conference on Mobile Syst., Appl., Services*. ACM, 2018, pp. 498–499.

[8] A. Juan-Verdejo and B. Surajbali, "Xaas multi-cloud marketplace architecture enacting the industry 4.0 concepts," in *Doctoral Conference on Computing, Electrical Industrial Syst.*. Springer, 2016, pp. 11–23.

[9] A. Bröring, S. Schmid, C.-K. Schindhelm, A. Khelil, S. Kabisch, D. Kramer, D. Le Phuoc, J. Mitic, D. Anicic, and E. Teniente López, "Enabling iot ecosystems through platform interoperability," *IEEE software*, vol. 34, no. 1, pp. 54–61, 2017.

[10] G. S. Ramachandran, R. Radhakrishnan, and B. Krishnamachari, "Towards a decentralized data marketplace for smart cities," in *4th IEEE Annual International Smart Cities Conference*. IEEE, 2018.

[11] K. R. Özyilmaz, M. Doğan, and A. Yurdakul, "Idmob: Iot data marketplace on blockchain," in *2018 Crypto Valley Conference on Blockchain Technol. (CVCBT)*. IEEE, 2018, pp. 11–19.

[12] M. Niekerk and R. Veer, "Global market for local data," *Whitepaper*, 2016.

[13] F. de la Vega, J. Soriano, M. Jimenez, and D. Lizcano, "A peer-to-peer architecture for distributed data monetization in fog computing scenarios," *Wireless Commun. Mobile Computing*, vol. 2018, 2018.

[14] P. Missier, S. Bajoudah, A. Capossele, A. Gaglione, and M. Nati, "Mind my value: a decentralized infrastructure for fair and trusted iot data trading," in *Proc. of the Seventh International Conference on the Internet of Things*. ACM, 2017, p. 15.

[15] D. Niyato, X. Lu, P. Wang, D. I. Kim, and Z. Han, "Economics of internet of things: An information market approach," *IEEE Wireless Commun.*, vol. 23, no. 4, pp. 136–145, 2016.

[16] Y. Zhang, Z. Xiong, D. Niyato, P. Wang, and Z. Han, "Market-oriented information trading in internet of things (iot) for smart cities," *arXiv preprint arXiv:1806.05583*, 2018.

[17] M. B. Safianowska, R. Gdowski, and C. Huang, "Revenue maximizing auction for perishable iot resources," in *Information Communication Technol. Convergence (ICTC), 2016 International Conference on*. IEEE, 2016, pp. 417–422.

[18] V. Charpenay, H. Nguyen, M. Ibrahim, A. Zappa, and A. Bröring, "Matching offerings and queries on an internet of things marketplace," in *European Semantic Web Conference*. Springer, 2018, pp. 66–71.

[19] E. I. P. Initiative, "IoT Marketplace," https://market.big-iot.org/, 2018, [Online; accessed 15-November-2018].

[20] A. S. Thuluva, A. Bröring, G. P. Medagoda, H. Don, D. Anicic, and J. Seeger, "Recipes for iot applications," in *Proc. of the Seventh International Conference on the Internet of Things*. ACM, 2017, p. 10.

[21] J.-S. Park, T.-Y. Youn, H.-B. Kim, K.-H. Rhee, and S.-U. Shin, "Smart contract-based review system for an iot data marketplace," *Sensors*, vol. 18, no. 10, p. 3577, 2018.

[22] K. Mišura and M. Žagar, "Data marketplace for internet of things," in *Smart Syst. Technol. (SST), International Conference on*. IEEE, 2016, pp. 255–260.

[23] Y. Zhang, Z. Xiong, D. Niyato, P. Wang, and J. Jin, "Joint optimization of information trading in internet of things (iot) market with externalities," in *Wireless Commun. Netw. Conference (WCNC), 2018 IEEE*. IEEE, 2018, pp. 1–6.

[24] A. Suliman, Z. Husain, M. Abououf, M. Alblooshi, and K. Salah, "Monetization of iot data using smart contracts," *IET Netw.*, 2018.

[25] T.-V. Nguyen, L.-S. Lê, H.-L. Truong, K. Nguyen-An, and P. H. Ha, "Handling service level agreements in iot= minding rules+ log analytics?" in *2018 IEEE 22nd International Enterprise Distrib. Object Computing Conference (EDOC)*. IEEE, 2018, pp. 145–153.

[26] A. Blázquez Rodríguez, "Security and aaa architectures in an iot marketplace," 2014.

[27] J. Mineraud, O. Mazhelis, X. Su, and S. Tarkoma, "A gap analysis of internet-of-things platforms," *Comput. Commun.*, vol. 89, pp. 5–16, 2016.

[28] S. Akkermans, N. Small, W. Joosen, and D. Hughes, "Niflheim: End-to-end middleware for applications across all tiers of the iot," in *Proc. of the 15th ACM Conference on Embedded Netw. Sensor Syst.*. ACM, 2017, p. 41.

[29] A. S. Al Rawahi, K. Lee, J. Robinson, and A. Lotfi, "Enabling exclusive shared access to cloud of things resources," in *Proc. of the 2018 Workshop on Theory Practice for Integrated Cloud, Fog Edge Computing Paradigms*. ACM, 2018, pp. 15–20.

[30] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[31] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[32] E. Ziztler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization," *Evolutionary Methods for Design, Optimization, Control*, pp. 95–100, 2002.

[33] D. A. Van Veldhuizen and G. B. Lamont, "Evolutionary computation and convergence to a pareto front," in *Late breaking papers at the genetic programming 1998 conference*, 1998, pp. 221–228.

[34] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Da Fonseca Grunert, "Performance assessment of multiobjective optimizers: An analysis and review," *TIK-Report*, vol. 139, 2002.

[35] R. H. Koochaksaraei, I. R. Meneghini, V. N. Coelho, and F. G. Guimarães, "A new visualization method in many-objective optimization with chord diagram and angular mapping," *Knowledge-Based Syst.*, vol. 138, pp. 134–154, 2017.

[36] Kai Zhao and Lina Ge. A survey on the internet of things security. In *2013 Ninth international conference on computational intelligence and security*, pages 663–667. IEEE, 2013.

Ahmed Salim Alrawahi (M'12) received his BSc. in Information Systems from University of Nizwa Oman in 2009. He received his MSc. in Network Management and Security from Murdoch University, Perth, Australia in 2012. He worked as a teaching assistant in Nizwa College of Applied Sciences, Nizwa Oman since then. Ahmed is currently pursuing his PhD in computer science at Nottingham Trent University, Nottingham, UK. His research interests include IoT, Cloud Computing and Optimisation approaches for combinatorial exchange problems.

Kevin Lee (SM'11) is a Senior Lecturer in Software Engineering and Internet of Things (IoT) at the School of Information Technology in Deakin University, Australia. Before joining Deakin University in 2018 he held the position of Senior Lecturer in Computing and Technology at Nottingham Trent University, UK (2015-2018). Previously, he held the position of Senior Lecturer in Information Technology at Murdoch University, Australia (2010-2015). He was a postdoctoral research fellow at the University of Mannheim, Germany (2009-2010). He was a research associate at the University of Manchester, UK (2006-2009). He received his PhD in Computer Science from Lancaster University, UK in 2006. He is currently also an appointed external examiner for the Department of Computing and Technology, Ulster University, for the Computing suite of degrees (2017-2018). He has over 70 publications in high quality journals and conferences, with a H-Index of 13 and over 1100 citations. His research has mainly been in the Distributed Systems and Autonomic Computing areas. He has published in the areas of IoT, Cloud Computing, Robotics, Embedded Systems, Big Data, P2P network monitoring and physiological computing. His current main active research area is in IoT and its integration with Cloud Computing in support of dynamic IoT applications.

Ahmad Lotfi (M'96-SM'08) received his BSc. and MTech. in control systems from Isfahan University of Technology, Iran and Indian Institute of Technology Delhi, India respectively. He received his PhD degree in Learning Fuzzy Systems from University of Queensland, Australia in 1995. He is currently a Professor of Computational Intelligence at Nottingham Trent University, where he is leading the research group in Computational Intelligence and Applications. His research focuses on the identification of progressive changes in behaviour of elderly people suffering from Dementia or any other cognitive impairments. Accurate identification of progressive changes through utilisation of unobtrusive sensor network or robotics platform will enable carers (formal and informal) to intervene when deemed necessary. Research collaboration is established with world-leading researchers. He has worked in collaboration with many healthcare commercial organisations and end-users including Tunstall Healthcare Group and Nottingham Adult Care. He has received external funding from Innovate UK, EU and industrial companies to support his research. He has authored and co-authored over 150 scientific papers in the area of computational intelligence, internet of things, abnormal behaviour recognition and ambient intelligence in highly prestigious journals and international conferences. He has been invited as an Expert Evaluator and Panel Member for many European and International Research Programmes.