A Multiobjective QoS Model for Trading Cloud of Things Resources

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

***Abstract*—The emerging Cloud of Things (CoT) paradigm promises to meet the diverse requirements of many real-world applications, which previously could not be fulfilled by either Cloud Computing or Internet of Things (IoT). Trading CoT resources is a challenging aspect particularly when managing Quality of Service (QoS) as resource providers and application developers have different priorities. This paper focuses on the challenge of supporting QoS when trading CoT resources and performing resource allocation. The contributions of this paper are 1) the problem of managing QoS while trading CoT resources is investigated as an optimisation problem 2) a QoS model is proposed to solve the problem by optimising five different QoS objectives 3) evaluation which confirms the feasibility of the proposed model in optimising diverse QoS requirements.**

***Index Terms*—Cloud Computing, Internet of Things, Cloud of Things, QoS, Trading, Optimisation, Resource Allocation**

I. Introduction

CLOUD Computing transforms computing resources into a modern utility. However, the physical scope of Cloud Computing is limited because it is focused on data-centres and does not interact with the environment. IoT aims to interconnect heterogeneous things that can interact with each other and the surroundings. The interaction of Cloud Computing and IoT overcomes the limited reach-ability of Cloud Computing and limited computational capabilities of IoT. A new computing paradigm called The Cloud of Things (CoT) [2] extends the limited scope of Cloud Computing and provides IoT with virtually unlimited resources [1].

Despite the strong interest in integrating Cloud Computing and IoT, there are still many open challenges [3]. One of these is in supporting Quality of Service (QoS) for CoT applications. All CoT applications focus on particular QoS attributes either explicitly or implicitly in the application aims. For example, latency sensitive applications (e.g. military, emergency services) benefit from the larger number of IoT sensing nodes. Less time-sensitive applications (e.g marketing, planning) utilise the scalability and reliability of the Cloud to process big data generated from distributed IoT resources and make decisions accordingly. Supporting QoS for these applications means enabling these attributes to be prioritised.

Supporting QoS in CoT applications is particularly challenging in scenarios where there are many resource providers and consumers such as in smart-cities. Using market-based mechanisms to commodify resources is an approach used in similar large-scale computing infrastructures such as Grids and federated Clouds. The commoditisation of CoT deployments will prevent the slow down in the rate of IoT adoption [4] caused by the considerable investment in hardware, software and maintenance. In a CoT marketplace, resources can be traded as commodities rather than as physical products and priced using Cloud pay-per-use pricing model. The commoditisation of CoT resources will reduce overall costs, enable sharing and reusing of IoT resources, and motivate for new services and applications. In this scenario, the use of resources will be very dynamic, and will require efficient market mechanisms to support QoS in CoT.

The aim of this research is to support QoS in the integration of Cloud and IoT. This is achieved by proposing an optimisation-based approach for managing QoS in trading CoT resources. The contributions of this paper are: 1) Investigating the problem of managing QoS in CoT by considering resource cost, response time, energy consumption, fault tolerance and resource coverage. 2) Proposing a new QoS model to optimise the QoS objectives as either a single-objective or multi-objective optimisation problem. 3) Performing rigorous simulations to evaluate the proposed model using three optimisation algorithms.

The remainder of this paper is as follows. In Section II, a review of the related work is presented. Section III describes the proposed QoS model and defines the problem of supporting QoS whilst trading resources in CoT. Evaluation results are discussed in Section IV. Conclusions and future work are presented in Section V.

II. Related Work

Many resource management problems in large-scale computing infrastructures are NP-hard [5]. This means there are no best or exact solutions to such problems in reasonable time due to the complexity, scalability and uncertainty of users' requirements. Similarly, CoT is a large-scale computing infrastructure by nature and its resource management aspects are challenging [6], [7].

A. Resource Allocation when Integrating Cloud and IoT

Resource allocation techniques in IoT environments are still emerging as part of other systems (e.g. Cloud Computing, CoT, WSNs). IoT Cloud approaches focus on integrating IoT resources with Cloud to enable on-demand provisioning of shared IoT resources via the Cloud of Things.

A. S. Alrawahi and A. Lotfi are with the School of Science and Technology, Nottingham Trent University, Nottingham, UK, NG11 8NS. E-mail: ahmed@alrawahi.org, 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

An early attempt to integrate wireless sensor networks (WSNs) and Cloud Computing has been discussed and implemented in [1]. The proposed architecture enables WSNs tasks to be offloaded to the Cloud. A scalable CoT architecture has been developed in [9] along with two algorithms to discover and virtualise IoT resources. The algorithms have been proposed to minimise the number of deployed resources and communication overhead. A three-tier CoT architecture has been proposed along with the development of multi-objective scheme to optimise task allocation in CoT [12]. The scheme aims to minimise energy consumption and latency. Another three-tier architecture is designed in [13] to enable sharing of Cloud resources in vehicular networks. The proposed system intends to reduce service dropping rate. Application-specific architectures are also proposed. An architecture that integrates sensors and Cloud Computing for military operations is presented in [15]. The proposed architecture allocates resources based on user prioritises to improve the performance and availability of resources.

Resource allocation in CoT is also addressed by various models and algorithms. A device/Cloud framework has been presented in [8] to enable collaboration between smart devices and Clouds. The framework uses real-world case studies to elaborate on the benefits of integrating smart devices and Cloud Computing. Another consensus-based framework has been presented in [11] to improve the lifetime of the connected resources when allocating them in the Cloud. A model for integrating sensors and Cloud Computing has been proposed in [10] to evaluate the cost-effectiveness and performance of a CoT architecture. A resource allocation algorithm is introduced in [14] to enable Cloud providers optimising the throughput, occupancy and utilisation of the IoT requests. A model has been developed in [16] to cooperate between airborne sensor network and back-end Cloud. The model applies heuristics to minimise the drones travel time and failures in meeting their deadlines.

B. *Commoditisation of CoT Resources*

A solution to the resource allocation problem in CoT is to enable efficient resource sharing. An obstacle to this is the lack of support in sharing CoT resources. An emerging trend is for market mechanisms to trade resources in large-scale infrastructures similar to CoT, such as Grids, Clouds, and Vehicular Networks [13], [17].

A model is proposed in [4] to create a trading-based value for IoT resources. It aims to enable sharing and reusing IoT resources by trading them similarly as Cloud resources. A marketplace architecture is designed in [18] to commodify and trade CoT resources. The trading problem is described as a multi-attribute combinatorial problem and vocabularies needed for the trading process are introduced. The development and implementation of a market-based model are presented in [19]. The three-tier model considers the Cloud as a broker for IoT resources. Resource allocation has been formulated as a multi-objective optimisation problem aiming to allocate traded resources with the minimum response time, minimum energy consumption and maximum profit for the broker.

A federation model for Cloud IoT providers is proposed in [20] to support market mechanisms. The model aims to satisfy providers' requirements and improve the rate of resource utilisation. An auction-based model is presented in [23] to assign CoT computation resources to the consumers which targets performance improvement when allocating distributed IoT resources. A reputation-based framework for CoT architectures is introduced in [22]. It employs an auction to select physical resources for sensing tasks and payments for users.

Market-based algorithms for CoT commoditisation are also investigated. A combinatorial auction algorithm is proposed in [21] to maximise the providers' profit and the rate of job completion. Another auction-based algorithm is developed in [24] to support resource allocation in CoT environments. The proposed algorithm aims to maximise the providers' profit while maintaining their capacity constraints.

C. *Quality of Service in Cloud of Things*

QoS is a description of the perceived performance of a particular service that can be tangible or non-tangible. To measure QoS in CoT, some attributes are chosen for evaluating the relative performance of a particular resource, service or application.

To support QoS in a new application domain is to define appropriate QoS attributes for that domain. In Cloud, there are Service Level Agreements (SLAs) targeted at QoS [25]. There have also been attempts at QoS in Cloud, with a particular focus on supporting different workloads and capacities [26]. Supporting QoS in virtualisation-based environments is a particularly challenging especially in trust and security-related issues [27].

For IoT, QoS-aware architecture presented in [28] with the focus on information collection and analysis of QoS aspects in the IoT system. Another architecture proposed in [29] to provide QoS-aware scheduling of IoT services. Further insights on QoS for IoT are discussed in [30].

D. *Gap Analysis*

QoS-aware resource allocation techniques have been studied for Cloud and IoT separately while they are still developing for the CoT [31], [32]. CoT is complex, with heterogeneous resources, which lends itself to the use of market-based mechanisms for achieving QoS-aware resource allocation. The approach is inspired by existing techniques used to allocate resources by trading them in similar large-scale environments including Cloud Computing and WSNs [17].

This paper intends to evaluate the use of optimisation algorithms when managing QoS in CoT environments. The approach of using optimisation algorithms to solve this trading problem is justified due to their capabilities in finding optimal solutions to similar problems in complexity and scalability. In this case, the complexity resides here due to the heterogeneity of Cloud and IoT resources that results in difficulties when quantifying their value and leading to the involvement of multifaceted variables and decisions.

III. QoS-Based Resource Allocation Model for CoT Applications

To support an efficient resource allocation for the emerging CoT applications, a generic and dynamic QoS model is needed. The QoS model is proposed here with the following assumptions/considerations. 1) CoT resources are allocated to the applications based on QoS attributes as part of a trading process where QoS is vital. The CoT applications are independent of each other. 2) The CoT application can simultaneously utilise multiple physical resources from different providers while maintaining the required QoS level collectively. 3) The CoT application should maintain a certain QoS level to fulfil consumers' requirements even in a case of conflicting ones at the same time (e.g. min. energy consumption, max. resource coverage).

A. QoS Attributes for CoT Application

The complex nature of CoT applications requires a generic QoS model to optimally allocate the required resources. The complexity resides here for two reasons. First, the heterogeneity of CoT resources makes it challenging to build a unified QoS model with a broad scope QoS attributes that can satisfy the QoS requirements of all applications. Second, CoT applications have divers QoS requirements that make it challenging to maintain the required QoS levels.

To overcome the above-mentioned obstacles, using optimisation strategies is considered to trade CoT resources while satisfying the QoS requirements. This approach supports a dynamic selection of QoS attributes based on the application requirements. Thus, allowing a better measurability of individual QoS attributes as discussed in Section III-F or collectively as presented in Section III-G. The main QoS attributes considered by this model are resource cost, resource coverage, response time, energy consumption and fault tolerance. A detailed description of each attribute is presented in Section III-F.

B. Problem Formulation

The QoS model assumes a CoT marketplace system M with multiple consumers $C = (c_1, ..., c_d)$ who request multiple set of resources $R = (r_1, ..., r_j)$ from multiple providers $P = (p_1, ..., p_m)$ to build multiple concurrent applications $A = (a_1, ..., a_z)$. Each application requires different QoS levels from others. Each provider deploys a set of heterogeneous resources while each consumer requests a set of homogeneous resources.

This model aims to optimally allocate resources to various applications while satisfying their QoS requirements. The resource allocation is considered optimal when it satisfies two conditions. First, the allocated resources to each application are sufficient to fulfil the minimal QoS requirements of the application. Second, the overall QoS objective for each application is maximised. A binary variable a_{ij} is presented in Equation 1 to describe the process of resource allocation.

$$a_{ij} = \begin{cases} 1, & \text{if } \ r_j \text{ is allocated to } a_i \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

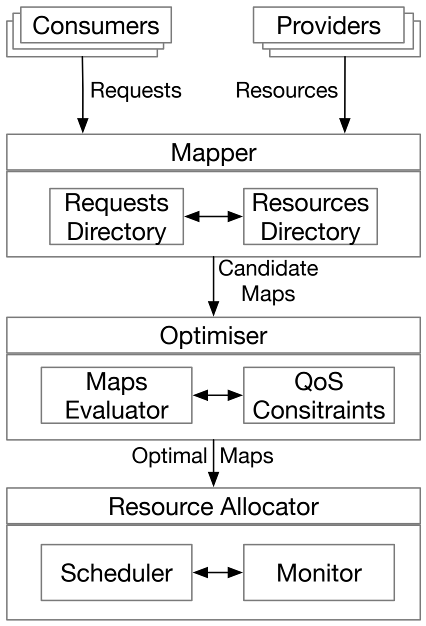


Fig. 1. High-level marketplace architecture

C. Problem Complexity

The resource allocation in large-scale computing infrastructures is described in the literature as NP-hard or NP-complete problem [19]. The complexity of allocating CoT resources with QoS constraints is described as follows.

The research space for the optimisation problem can be formed by considering the total number of requests RQ, the number of available resources to match these requests R and the number of resources that violates the QoS constraints V. This can be formulated as $RQ^R - V$. To illustrate, if we consider $RQ = 10$ and $R = 10$ without any constraints, the search space is formed of 10 billion possible solutions represented as 10^{10}.

There are two conflicting considerations here. First, violations of constraints are expected to exist which limit the search space and consequently the problem complexity. Second, the violations of the QoS constraints is not expected to reduce neither the size of the search space nor the problem complexity due to the heterogeneity of CoT resources and the scalability of the problem. A CoT marketplace is expected to host a large number of heterogeneous resources that increases the search space exponentially. To relax this challenge, the QoS attributes are considered as utility functions to be optimised individually as a single objective problem or collectively as a multi-objective problem. The following sections discuss each utility in details.

D. Marketplace System Architecture

For efficient resource allocation with QoS support in CoT, efficient commiditisation of CoT resources has to be enabled. To achieve this goal, a marketplace system architecture is depicted in Figure 1. The system architecture and the process of finding an optimal resource allocation solution are described as follows.

Consumers submit their application requests and providers submit their resource offerings to the marketplace. Requests and resources are stored in different directories where the mapper can generate candidate maps of mapped resources to applications. The mapper transfers candidate maps to the optimiser for QoS evaluation. In the optimiser component, the evaluator assesses candidate maps based on the QoS constraints available for each round of the optimisation cycle. The evaluator terminates its cycle when the optimal map is found. The resource allocator is responsible for the overall resource allocation process. The scheduler maintains the resources and applications schedules where it controls the lease-time of resources and manages the allocated resources in the Cloud. The allocator also orchestrates the process of joining and dis-joining resources based on the proposed schedules. The monitor component communicates the resource allocation events with the system, consumers and providers.

The use of the optimisation component provides significant flexibility to this approach. It can be implemented as a core of system architecture or as a complementary to other market-based mechanisms. When used as part of system architecture, it can be adopted as a substitute for the core component in one of the following market structures. 1) broker system, 2) monopoly market, 3) oligopoly market, 4) single-side auction and 5) double-side auction. This marketplace system aims to satisfy the market structure of double-sided auction.

E. Illustrative Scenario

To elaborate, the following scenario presents a use case of QoS-driven resource allocation using the marketplace system. A high-density metropolitan area is considered a desirable location for multiple public, private and academic organisations to implement their IoT environmental monitoring applications. Applications monitor various indicators including light, pollution, temperature, pressure, humidity and wind. Considering the existing IoT practice, each organisation is required to deploy its infrastructure (e.g. various sensors, dedicated network or gateways to the Internet, other computing nodes) and develop its application. This may not be feasible for all interested parties or expensive replication is created otherwise.

The proposed optimisation-based approach separates between infrastructure deployment and application development. Providers can deploy their resources across the metropolitan area and submit their offerings to the marketplace. Consumers also submit their applications requirements to the marketplace to match them with the required resources. The mapping process is based on the QoS requirements of applications. As these applications are financially constrained, public and academic organisations can prioritise their requests with minimised cost and energy consumption while private organisations can prioritise their requests with maximised area coverage and fault tolerance. Upon successful resource allocation, each application can send a software component (e.g. Java applet or Python script) to configure and utilise the acquired resources based on their application and QoS requirements.

F. Single Objective Optimisation Problem

Objective 1: Minimising Cost. Consumers aim to have a cost-efficient resource allocation. The cost of resources is an important aspect to be considered while optimising QoS levels. The importance comes from the balance enforced by the cost when other QoS constraints exist. To elaborate, an application requires a certain level of response time, energy consumption and fault tolerance within a limited budget constraint of the consumer. Without considering the cost as a constraint, there would be more resource allocation options for the application where many of them are not feasible.

To minimise the cost of allocated resources, let cs_j be the resource cost whereas the consumer bid is set to be b_i. A utilisation time t_i is specified by consumers RQ to set the time bounds. TQ_{ij} donates the estimated transmission and delay time that can impact the total utilisation time. TQ_{ij} consists of T_{ij} which is the latency between consumer i and provider j while dl_{ij} is the distance between a requested resource from consumer i and the actual location of the resource from provider j. TQ_{ij} is measured by $TQ_{ij} = \frac{T_{ij}}{dl_{ij}}$. Let rp_j donates the reputation of the provider based on the credibility measures of the marketplace. rp_j determines the trustworthiness of the provider and indicates a correlation with the ability of meeting applications' QoS requirements. To optimise the cost utility, the following objective is formulated.

$$\text{Minimise} \quad CS = \sum_{i=1}^{n}\sum_{j=1}^{m}(b_i - cs_j)t_i + (t_i TQ_{ij})rp_j \tag{2}$$

$$\text{subject to} \quad \sum cp_i \le cp_j \tag{3}$$

$$0 < cs_j \le b_i \tag{4}$$

$$0 < Er_i \le Ep_j \tag{5}$$

$$se_i \le se_j, \forall se \in SE \tag{6}$$

$$dl_{ij} \le Cv_j, \forall Cv \in CV \tag{7}$$

$$rp_i \le rp_j, \forall rp \in \tag{8}$$

$$ra_i \le ra_j, \forall ra \in RA \tag{9}$$

where i=1,...,n and j = 1,...,m for constraints 3, 4, 5, 6, 7, 8, 9.

Optimisation constraints provide significant support to the proposed model where additional measures can be formulated to enforce the QoS requirements. Constraint 3 limits the resource allocation to the capacity of providers and ensures the fair distribution of resources from multiple providers. cp_i is set to the number of requests from a consumer whereas cp_j donates the capacity of a provider. Thus, the number of requests does not exceed the capacity limit. Constraint 4 indicates whether both the cost of a resource cs_j and the bid from a consumer b_i are always positive and b_i has to be always greater than cs_j.

Constraint 5 presents an energy consumption constraint in which the required energy Er_i for an application does not exceed the available resource energy Ep_j. Zero or negative values of Ep_j indicates the unavailability of the resource due to power lifetime. Constraint 6 ensures the security requirements of the application se_i can be satisfied by the security capabilities of the resource se_j. Constraint 7 illustrates a constraint to ensure the maximum acceptable distance between the required coverage area of an application dl_{ij} is within the boundaries of the allocated resource coverage Cv_j.

To address the challenges of provider credibility, Constraint 8 ensures each provider maintains the minimal credibility requirements to formulate a reputation rate rp_j in the marketplace. The constraint also assures the minimal required reputation level rp_i of a consumer is met. Constraint 9 specifies the bounds for a set of attributes. The set aims to identify the hardware properties of the physical CoT resource that impact QoS directly or indirectly. This includes specifications of the processing, storage, memory, actuating and sensing components of the CoT nodes. Each property can expand into a multilevel sub-properties to improve the optimality of the resource allocation. For instance, the sensing component(s) of a resource described by its properties [*sensorType = [footfall, environmental, light], sensingRange = [0: poor, 1: good, 2: very good, 3: excellent], sensorAccuracy= 0: poor, 1: good, 2: very good, 3: excellent*] and so on. The resource attribute constraint offers the flexibility required for trading heterogeneous resources where QoS would significantly vary without a genuine approach of defining the QoS requirements/levels.

Objective 2: Minimising Response Time. Response time is an important QoS consideration, especially in large-scale distributed systems. CoT can be very widely distributed across a large geographical area where the response time is vital for application QoS. Latency is one contributor to response time. Variable L_{ij} corresponds to the latency between a consumer and a provider and it is measured by $L_{ij} = t_{ack} - t_{start}$. This measures the elapsed time from submitting the request by consumer t_{start} to the time of receiving an acknowledgement from a provider t_{ack}. The R_t utility also consider the estimated queuing and transmitting delays t_{qd} that is expected to be at its minimal for many time-sensitive applications. It is calculated as $t_{qd} = \frac{(L_{ij})}{dl_{ij}}$ where dl_{ij} is the distance between the consumer and the provider. R_t utility can be optimised as follows.

$$\text{Minimise} \quad R_t = \sum_{i=1}^{n}\sum_{j=1}^{m} L_{ij} + t_{qd} \tag{10}$$

$$\text{subject to} \quad 3, 4, 5, 6, 7, 8, 9 \tag{11}$$

Objective 3: Minimising Energy Consumption. The energy efficiency is a critical measurement for QoS in CoT application. Many IoT physical resources are power-constrained in which their performance are limited. The energy consumption utility E aims to minimise the power consumption of allocated resources while being utilised by consumers. This can be presented by the difference between the initial power supply of the resource Ep_j and the estimated power consumption requested by the consumer Er_i. This can be optimised as follows.

$$\text{Minimise} \quad E = \sum_{i=1}^{n}\sum_{j=1}^{m} Ep_j - Er_i \tag{12}$$

$$\text{subject to} \quad 3, 4, 5, 6, 7, 8, 9 \tag{13}$$

Objective 4: Maximising Fault Tolerance. Fault tolerance in this context describes the ability of a set of allocated resources to continue providing an acceptable service level in case of a failure. The proposed QoS model in this study considers both soft and hard faults for IoT resources.

The use of concurrent communication interfaces in a resource is donated by mu_j. This enables allocated resources to reconfigure a different interface for the same application in which resources were assigned to. In case of unavailability of multiple interfaces in a resource $mu_j = 0$, the providers may already have deployed a redundant or standby resources rr_j nearby with the similar QoS attributes of the failing resource. Another important aspect that may impact the recovery of a resource from failures is the difference in response time of that resource during or after a failure. The variable ΔRt donates the difference between the current response time after failure βRt and the average Rt where $\Delta Rt = \beta Rt - avg(Rt)$. In order to optimise the fault tolerance utility, the following objective function is presented.

$$\text{Maximise} \quad F_t = \sum_{i=1}^{n}\sum_{j=1}^{m} mu_j + rr_j - \Delta Rt \tag{14}$$

$$\text{subject to} \quad 0 \le mu_i \le mu_j \tag{15}$$

$$cr_i \le cr_j, \forall cr \in CR \tag{16}$$

$$rr_i \le rr_j, \forall rr \in RR \tag{17}$$

$$3, 4, 5, 6, 7, 8, 9 \tag{18}$$

where i=1,...,n and j = 1,...,m for constraints 16, 17 and 18.

Due to the vitality of fault tolerance for QoS requirements, the following constraints are enforced. Constraint 16 indicates whether a resource supports concurrent interfaces or not where $mu_j = 0$ means the resource has one interface only. The constraint also assures that the minimal number of requested interfaces mu_i is satisfied. Constraint 17 is set to minimise the impact of communication reliability during failures. Let cr_i the required level of communication reliability for an application while cr_i is the actual communication reliability of the allocated resource to that application. Constraint 18 ensures the required level of redundancy by an application rr_i can be satisfied by the correspondent level of the provider rr_j.

Objective 5: Maximising Resource Coverage. Many CoT applications require a specific area coverage, especially for sensing capabilities. Without certain coverage level, CoT applications may not achieve their reach-ability goals. The proposed QoS model considers the resource coverage as an integral QoS utility for CoT applications. The resource coverage can be calculated using the sensing range s_j of the resource and the maximum transmission power Et_{max} available. The distance dl_{ij} between requested location and the actual location of the resource is also considered. To optimise the resource coverage, the following objective is formulated.

$$\text{Maximise} \quad Cv = \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{s_j \times Et_{max}}{dl_{ij}} \tag{19}$$

$$\text{subject to} \quad 3, 4, 5, 6, 7, 8, 9 \tag{20}$$

G. Multiobjective Optimisation Problem

In Section III-F, the QoS attributes are presented as individual objectives. In a marketplace environment, consumers are expected to have a multi-attribute QoS for their CoT applications. This adds considerable complexity to the problem due to the following reason. QoS attributes may conflict with one another in which trade-offs between conflicting attributes has to be taken into account. For instance, an application requires a set of resources with minimum cost, response time and the maximum possible area coverage. To overcome this challenge, the proposed QoS utilities are re-defined as a multi-objective optimisation problem as follows.

$$\text{Minimise} \quad CS = \sum_{i=1}^{n}\sum_{j=1}^{m}(b_i - cs_j)t_i + (t_i TQ_{ij})rp_j \tag{21}$$

$$\text{Minimise} \quad R_t = \sum_{i=1}^{n}\sum_{j=1}^{m} L_{ij} + t_{qd} \tag{22}$$

$$\text{Minimise} \quad E = \sum_{i=1}^{n}\sum_{j=1}^{m} Ep_j - Er_i \tag{23}$$

$$\text{Maximise} \quad Cv = \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{s_j \times Et_{max}}{dl_{ij}} \tag{24}$$

$$\text{Maximise} \quad F_t = \sum_{i=1}^{n}\sum_{j=1}^{m} mu_j + cr_j + rr_j - \Delta Rt \tag{25}$$

$$\text{subject to} \quad 3, 4, 5, 6, 7, 8, 9, 16, 17, 18 \tag{26}$$

To solve the problem of resource allocation with QoS constraints, the following optimisation algorithms are used. The improved Strength Pareto Evolutionary Approach (SPEA2) [33], A Multiobjective Evolutionary Algorithm Based on Decomposition (MOEA/D) [34] and Multi-Objective Indicator-Based Evolutionary Algorithm (IBEA) [35]. These algorithms are chosen for the following three reasons. First, they are gradient-free strategies. This means derivatives calculation is not required which can be computationally expensive. Second, using derivatives-free algorithms gives the advantage of avoiding local optima solutions in many cases. Third, these algorithms are known to solve problems similar to trading CoT resources in complexity and scalability.

IV. Evaluation

This section presents the experimental setup, analyses and discusses the results of resource allocation with five different QoS utilities.

A. Experimental Setup

The simulated marketplace system [32] is assumed to use different optimisation strategies to map the optimal resources that satisfy the QoS requirements of multiple CoT applications. The participants of this simulation are summarised in Table I and described as follows. 10 consumers submit a total number of 10K requests to the marketplace where a number of 20 providers offers 200K heterogeneous resources deployed in a circle area of 2000 meter radius. Each consumer is assumed to request a homogeneous type of resources to be allocated for one application. Experiments presented in this section has the following aims. First, to assess the feasibility and practicability of the proposed QoS model for CoT applications. Second, to evaluate the performance of different optimisation strategies when optimising QoS-based utilities.

TABLE I
Simulation parameters.

Parameter	**Value**
Simulated Area Radius	2 Km
Number of Requests	10K
Number of Resources	200K
Number of Consumers	10
Number of Providers	20
Number of Applications	10

Algorithm	Parameters
SPEA2 generated between 1 and len(requests)	Indicator value $K = 1$, initial population rando
SPEA2	Indicator value $K = 1$, initial population rando

Experiments using a synthetic data-set in this study is justified as follows. First, it is technically challenging and financially unfeasible to build a real test-bed for this problem with similar scalability to a real-world scenario. Second, to the best of our knowledge, there is no public available meta-data of IoT physical resources that can be used to implement the proposed QoS model. To overcome both challenges, a large set of meta-data for 200k resources is generated based on the properties of IoT nodes surveyed from several IoT vendors including Amazon, Microsoft and Google.

The experimental environment is Python 3.6 for 64-bit Mac OS with a 2.6 GHz Intel Core i7 processor and a 16 GB RAM. The common parameters are a maximum number of 200 iterations with a population size of 200. The algorithm-specific parameters are described in Table II.

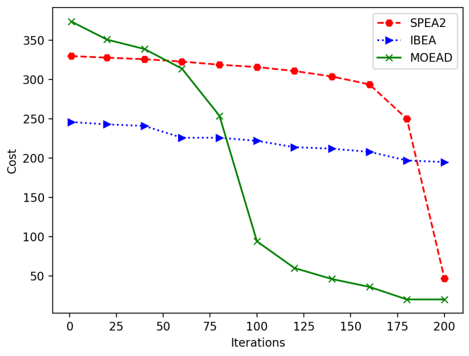
B. Experimental Results

As discussed earlier in Section III-B, the problem of resource allocation with QoS constraints is defined as a single objective optimisation problem where the QoS utility functions are optimised individually and also defined as a multiobjective optimisation problem where the QoS utility functions are optimised collectively. In this section, two categories of results are presented as follows.

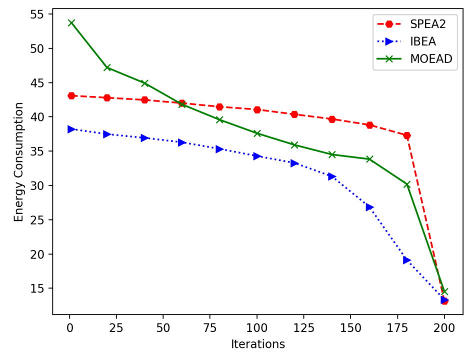
1) Single Objective Problem: To evaluate the proposed QoS objectives, each algorithm is run to optimise each individual QoS utility individually. Figures 2a, 2b and 2c illustrate the optimal resource allocation solutions for the cost utility, energy consumption and the response time at the end of each iteration, respectively. The results show that there is no dominant algorithm across the three utilities. Figure 2a shows SPEA2 and MOEAD algorithms converged to optimal solutions while IBEA is trapped into local minima. Figure 2b compares the

TABLE II
Algorithm-specific parameters.

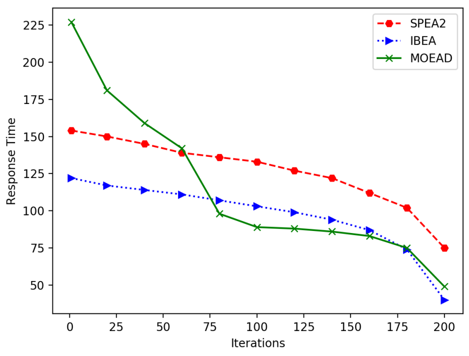
Algorithm	Parameter
SPEA2	Indicator value $K = 1$, initial population randomly generated between 1 an
IBEA	Initial population randomly generated between 1 and len(requests)
MOEAD	Neighbourhood size = 200,initial population randomly generated between 1



(a)

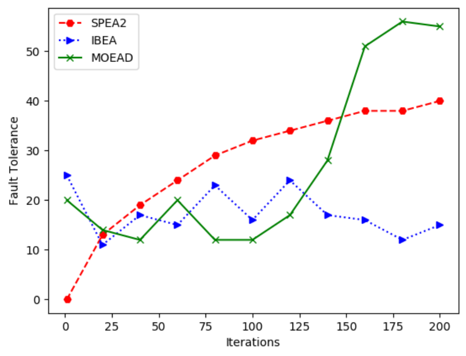


(b)

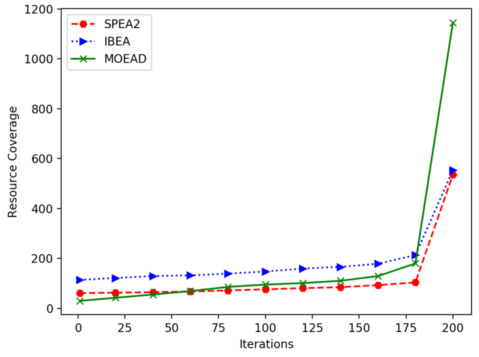


(c)

Fig. 2. Results of minimising different utilities (a) Cost of resources (b) Energy Consumption (c) Response time.



(a)



(b)

Fig. 3. Results of maximising different utilities (a) Fault tolerance (b) Resource coverage.

minimisation of energy consumption. All algorithms have a different start but converged towards the same solution at the last iteration. The results of minimising the response time utility are illustrated in Figure 2c where all algorithms could find a different optimal solution with notable out-performance of IBEA and MOEAD when compared with SPEA2.

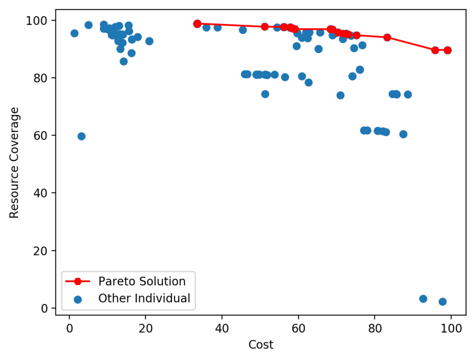
Figures 3a, 3b present illustrative comparisons of the algorithms when maximising the fault tolerance and the resource coverage utilities, respectively. In Figure 3a, IBEA falls again into the local maxima which implies its inability to avoid local optima or to recover once falls into one. Both SPEA2 and MOEAD find different optimal allocation assignments where MOEAD significantly outperformed SPEA2. Figure 3b compares between the optimisers when maximising the resource coverage utility. It is clear that the performance of MOEAD is far better than the other two.

From results compared in the above-mentioned figures, the following can be observed. First, there are at least two optimal solutions for each QoS utility. Second, MOEAD contributes to the optimality of cost, fault tolerance and the resource coverage more than SPEA2 and IBEA.

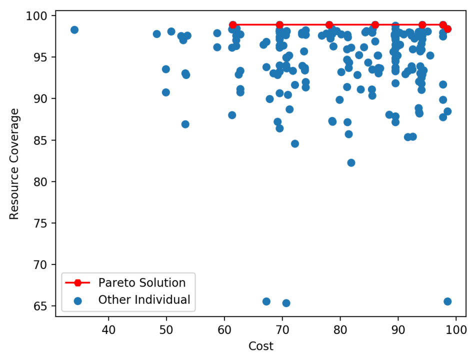
2) Multi-Objective Problem: As discussed earlier, performing a multiobjective optimisation is necessary to address the QoS requirements of applications when trading CoT resources.

The five QoS objectives are described as a multi-objective optimisation problem and solved by the three algorithms accordingly. The multi-objective will yield different optimal solutions rather than a single solution. The optimal solutions are called a Pareto Front and a decision has to be made to select the best solution. In CoT marketplace, it is assumed that the decision is made autonomously by the marketplace system based on predefined preferences of a consumer.

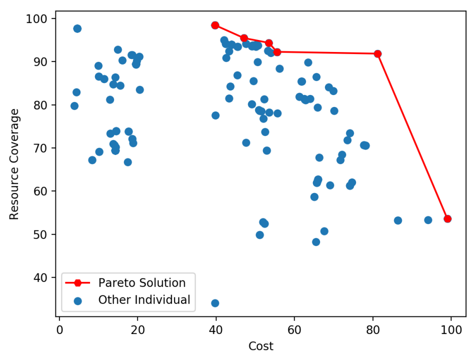
The results presented in Figures 4, 5, 6, 7, 8, 9 show bi-objective optimisation of real-world QoS requirements. This includes minimising the cost while maximising the resource coverage, minimising the cost while maximising the fault tolerance, minimising the cost and the response time, minimising the energy consumption while maximising the resource coverage, minimising the energy and response time and max-



(a)

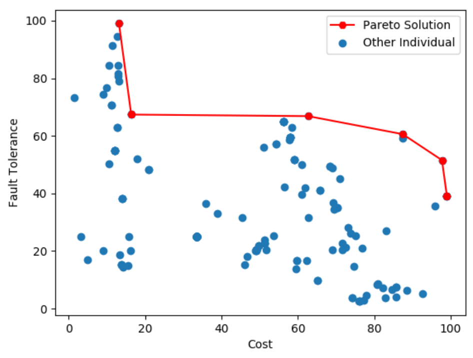


(b)

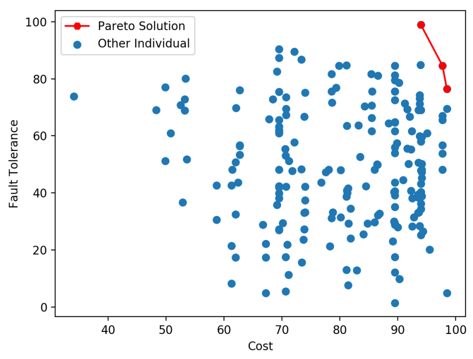


(c)

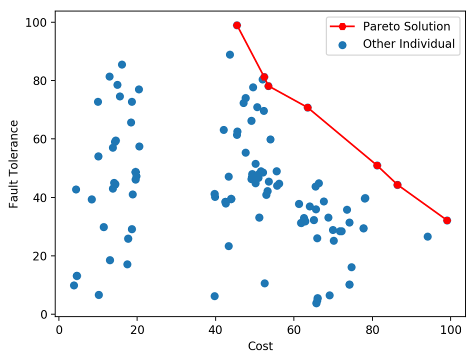
Fig. 4. Pareto optimal results minimising the cost while maximising the resource coverage (a) IPEAT algorithm (b) SPEAT algorithm (c) MOEAD algorithm.



(a)



(b)



(c)

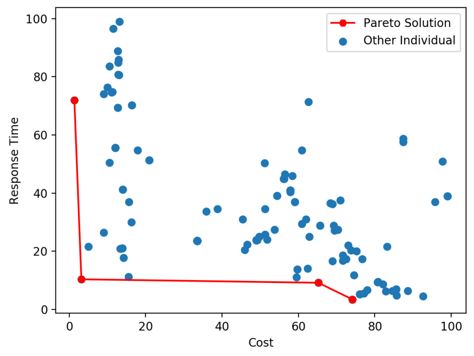
Fig. 5. Pareto fronts of minimising the cost while maximising the fault tolerance (a) IPEAT algorithm (b) SPEAT algorithm (c) MOEAD algorithm.

imising fault tolerance and resource coverage, respectively. Figure 4 illustrates the various optimal resource allocation maps that minimise the costs and maximise the resource coverage. Figures 4a, 4c show that an optimal solution of cost about 40 while the resource coverage is maximised to approximately 100. SPEAT algorithm produces slightly similar resource coverage as shown in Figure 4b but provide a bit higher cost.

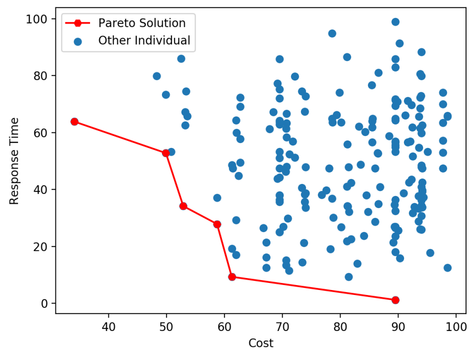
The Pareto fronts of minimising the cost while maximising the fault tolerance are presented in Figure 5. The Pareto fronts of IPEAT algorithm illustrated in Figure 5a show several fronts including two which far outperform the other two algorithms, especially in minimising the cost.

IPEAT and MOEAD algorithms compete to minimise the cost and response time as demonstrated in Figure 6. SPEAT produces one optimal solution that optimises the response time well but provides unbalanced cost to response time fronts which may not be attractive for consumers especially with time-sensitive applications.

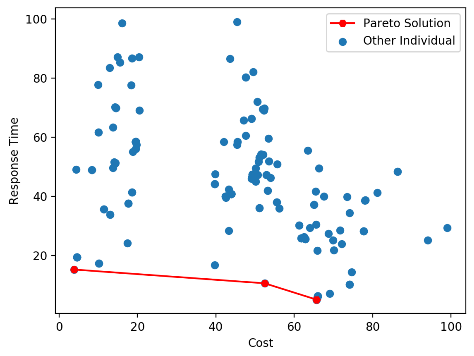
Figure 8 corresponds to applications that require minimising energy and response time. All algorithms presented compete well and minimise their fronts to the near-optimal solutions. Figure 8b shows SPEAT with only one front that represents a solution near zero for both axes. Two near-optimal solutions are presented in Figure 8c where response time and energy



(a)

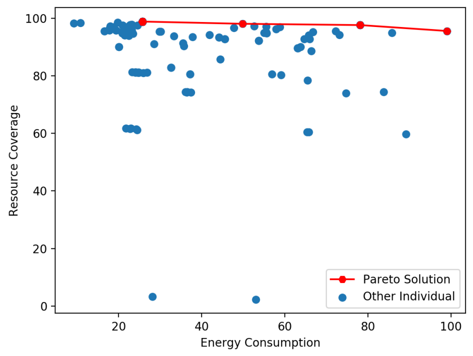


(b)

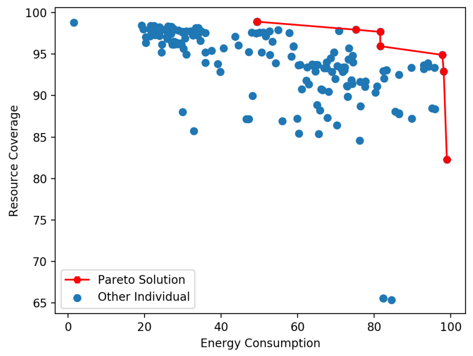


(c)

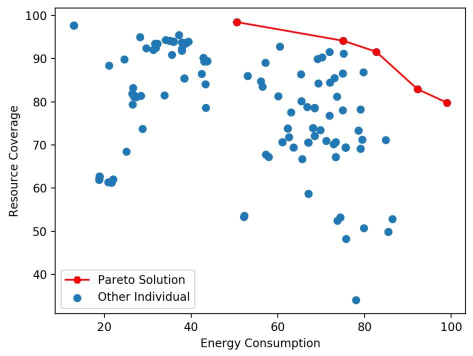
Fig. 6. Pareto optimal results minimising the cost and the response time (a) IPEAT algorithm (b) SPEAT algorithm (c) MOEAD algorithm.



(a)



(b)



(c)

Fig. 7. Pareto optimal results minimising the energy consumption and maximising the resource coverage (a) IPEAT algorithm (b) SPEAT algorithm (c) MOEAD algorithm.

consumption do not exceed 20. IPEAT algorithm produces the largest set of Pareto fronts in this scenario as shown in Figure 8a. All fronts have a response time less than 20 with reasonable energy consumption.

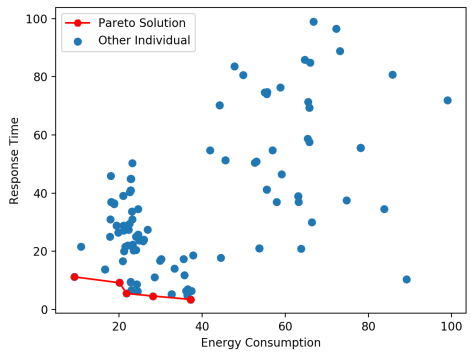
In Figure 9, Pareto optimal results maximising fault tolerance and maximising resource are illustrated. All algorithms produce at least one or more optimal front near 100 for the resource coverage and fault tolerance alike.

Table III summarises and compares the results produced by performing the optimisation of five QoS utilities. The table includes the minimum, average and maximum solution for each algorithm and utility.

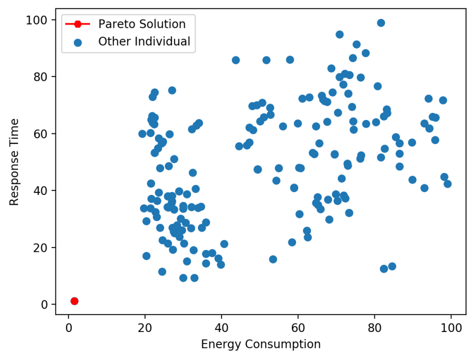
C. Discussion

Resource allocation in CoT marketplace is described as a single-objective and multi-objective optimisation problem. The simulation results show that the approach used in this study is practical for allocating resources to applications with QoS requirements. Results also show the ability of optimisers to produce at least one optimal solution for each individual utility tested. This may imply that Optimisation strategies can be used as a market mechanism for trading CoT resources instead of using traditional auctioneers.

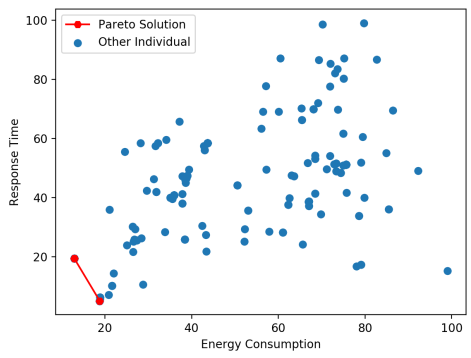
The proposed QoS model is architecture-independent and can be implemented by any marketplace system. It can also



(a)

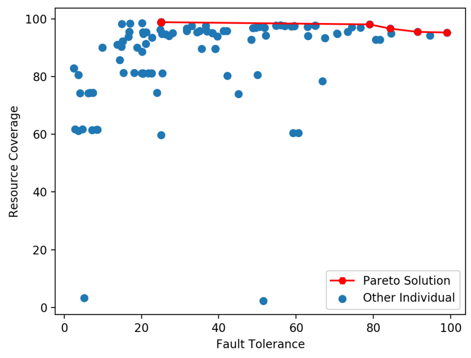


(b)

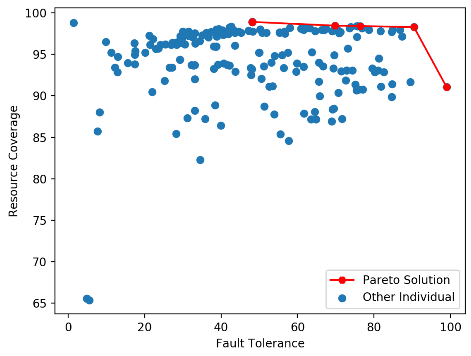


(c)

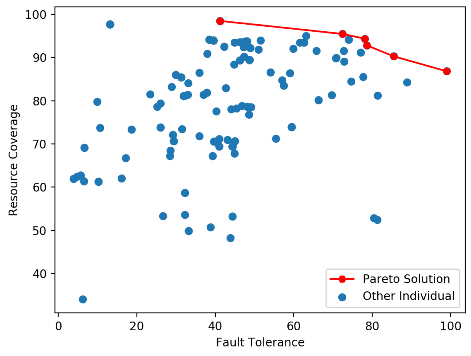
Fig. 8. Pareto optimal fronts minimising the energy consumption and the response time (a) IPEAT algorithm (b) SPEAT algorithm (c) MOEAD algorithm.



(a)



(b)



(c)

Fig. 9. Pareto optimal results maximising fault tolerance and resource coverage (a) IPEAT algorithm (b) SPEAT algorithm (c) MOEAD algorithm

be implemented as a complementary trading mechanism to support other trading mechanisms. This support separating the development of CoT applications from the deployment of physical resources making it easy to add any QoS objectives. Utility functions used with vocabularies proposed show their effectiveness in quantifying the value of various CoT resources. This implies potential higher satisfaction for the QoS requirements.

Implementation challenges are summarised as follows. 1) Optimisation algorithms fall into local optima (minima and maxima) which may not be preventable in some situations. 2) High memory utilisation is observed during the run of

TABLE III
SIMULATION RESULTS COMPARISON.

Algorithm(Utility)	Min	Avg	Max
SPEAT($Cost$)	34.09	80.20	98.45
SPEAT(E)	1.56	50.91	98.99
SPEAT(R_t)	1.25	48.93	98.99
SPEAT(Cv)	65.37	94.36	98.90
SPEAT(F_t)	1.42	49.55	98.99
IPEAT($Cost$)	1.34	44.61	98.99
IPEAT(E)	9.21	34.17	98.99
IPEAT(R_t)	3.42	31.44	98.99
IPEAT(F_t)	2.50	33.53	98.99
IPEAT(Cv)	2.31	91.48	98.85
MOEAD($Cost$)	3.84	40.49	98.99
MOEAD(E)	12.95	51.26	98.99
MOEAD(R_t)	5.14	44.93	98.99
MOEAD(F_t)	3.87	41.80	98.99
MOEAD(Cv)	34.09	80.20	98.45

experiments.

V. Conclusions and Future Work

Managing QoS in CoT environments is challenging. This challenge is relaxed by defining the problem of resource allocation in CoT trading setup as a single objective and multi-objective optimisation problem to satisfy several QoS requirements. Using different optimisation algorithms as a market mechanism is the approach considered to evaluate the proposed QoS model. Three optimisation strategies are applied to optimise QoS utilities including consumer cost, response time, energy consumption, area coverage and fault tolerance.

Simulation results confirm the practicability of trading heterogeneous CoT resources from multiple providers and consumed by multiple consumers. Using QoS utilities and proposed notations support quantifying the value of CoT resources. Pareto fronts are used to provide different optimal solutions for the utility functions.

Future work will take into account the following. First, assessing the scalability of this approach by optimising larger sets of resources. Second, optimising more QoS utilities to address application-specific requirements. Third, implementing this approach using different optimisation strategies.

Acknowledgement

The authors would like to thank Ministry of Higher Education, Oman, for the scholarship number (PGE023347) that enabled the first author to work on this project.

References

[1] K. Lee, D. Murray, D. Hughes, and W. Joosen, "Extending sensor networks into the cloud using amazon web services," in *Netw. Embedded Syst. Enterprise Appl. (NESEA), 2010 IEEE Int. Conf.* 2010, pp. 1–7.

[2] 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 *Applied Sci. and Technology (IBCAST), 2014 11th Int. Bhurban Conf.*. IEEE, 2014, pp. 414–419.

[3] M. Díaz, C. Martín, and B. Rubio, "State-of-the-art, challenges, and open issues in the integration of internet of things and cloud computing," *J. Netw. Comput. Appl.*, vol. 67, pp. 99–117, 2016.

[4] C. Perera and A. Zaslavsky, "Improve the sustainability of internet of things through trading-based value creation," in *Internet of Things (WF-IoT), 2014 IEEE World Forum.* 2014, pp. 135–140.

[5] M. Guzek, P. Bouvry, and E.-G. Talbi, "A survey of evolutionary computation for resource management of processing in cloud computing," *IEEE Comput. Intell. Mag.*, vol. 10, no. 2, pp. 53–67, 2015.

[6] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *Commun. Surveys & Tuts.*, vol. 17, no. 4, pp. 2347–2376, 2015.

[7] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications," *IEEE IoTJ*, vol. 4, pp. 1125–1142, 2017.

[8] Y. Yoon, D. Ban, S. Han, D. An, and E. Heo, "Device/cloud collaboration framework for intelligence applications," in *IoT*. Elsevier, 2016, pp. 49–60.

[9] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "Cloud of things for sensing as a service: sensing resource discovery and virtualization," in *Commun. Conf. (GLOBECOM),IEEE.* 2015, pp. 1–7.

[10] S. Misra, S. Chatterjee, and M. S. Obaidat, "On theoretical modeling of sensor cloud: A paradigm shift from wireless sensor network," *IEEE Syst. J.*, vol. 11, no. 2, pp. 1084–1093, 2017.

[11] V. Pilloni and L. Atzori, "Consensus-based resource allocation among objects in the internet of things," *Ann. Telecom.*, vol. 72, no. 7-8, pp. 415–429, 2017.

[12] W. Li, I. Santos, F. C. Delicato, P. F. Pires, L. Pirmez, W. Wei, H. Song, A. Zomaya, and S. Khan, "System modelling and performance evaluation of a three-tier cloud of things," *Future Generation Comput. Syst.*, vol. 70, pp. 104–125, 2017.

[13] R. Yu, Y. Zhang, S. Gjessing, W. Xia, and K. Yang, "Toward cloud-based vehicular networks with efficient resource management," *IEEE Netw.*, vol. 27, no. 5, pp. 48–55, 2013.

[14] H. S. Narman, M. S. Hossain, M. Atiquzzaman, and H. Shen, "Scheduling internet of things applications in cloud computing," *Ann. Telecom.*, vol. 72, no. 1-2, pp. 79–93, 2017.

[15] S. Misra, A. Singh, S. Chatterjee, and M. S. Obaidat, "Mils-cloud: A sensor-cloud-based architecture for the integration of military tri-services operations and decision making," *IEEE Syst. J.*, pp. 628–636, 2016.

[16] E. Tuyishimire, I. Adiel, S. Rekhis, B. A. Bagula, and N. Boudriga, "Internet of things in motion: A cooperative data muling model under revisit constraints," in *UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 2016 Intl. Conf.*. IEEE, 2016, pp. 1123–1130.

[17] A. S. Alrawahi and K. Lee, "Multi-attribute combinatorial marketplaces for cloud resource trading,"*Cloud and Green Comput. (CGC), 2nd Int. Conf.* IEEE, 2012, pp. 81–88.

[18] A. S. Alrawahi, K. Lee, and A. Lotfi, "Trading of cloud of things resources," *Proc. 2nd Int. Conf. IoT and Cloud Computing*, ACM, 2017, pp. 163:1–163:7.

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

[20] 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," *Fut. Gen. Comput. Syst.*, vol. 70, pp. 126–137, 2017.

[21] Y. Choi and Y. Lim, "Optimization approach for resource allocation on cloud computing for iot," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 3, p. 3479247, 2016.

[22] B. Kantarci and H. T. Mouftah, "Trustworthy sensing for public safety in cloud-centric internet of things," *IEEE IoTJ*, vol. 1, no. 4, 2014.

[23] S. Kim, "Nested game-based computation offloading scheme for mobile cloud iot systems," *EURASIP J. Wireless Commun. and Netw.*, vol., no. 1, p. 229, 2015.

[24] A. Gharaibeh, A. Khreishah, M. Mohammadi, A. Al-Fuqaha, I. Khalil, and A. Rayes, "Online auction of cloud resources in support of the internet of things," *IEEE IoTJ*, vol. 4, no. 5, pp. 1583–1596, 2017.

[25] D. Serrano, S. Bouchenak, Y. Kouki, T. Ledoux, J. Lejeune, J. Sopena, L. Arantes, and P. Sens, "Towards qos-oriented sla guarantees for online cloud services,"*Cluster, Cloud and Grid Computing (CCGrid), 13th IEEE/ACM Int. Symposium.* IEEE, 2013, pp. 50–57.

[26] D. Ardagna, G. Casale, M. Ciavotta, J. F. Pérez, and W. Wang, "Quality-of-service in cloud computing: modeling techniques and their applications," *J. Internet Services and Appl.*, vol. 5, no. 1, p. 11, 2014.

[27] P. Manuel, "A trust model of cloud computing based on quality of service," *Ann. Operations Research*, vol. 233, no. 1, pp. 281–292, 2015.

[28] R. Duan, X. Chen, and T. Xing, "A qos architecture for iot," *IoT, 4th Int. Conf. Cyber, Physical Social Comput.*. IEEE, 2011, pp. 717–720.

[29] L. Li, S. Li, and S. Zhao, "Qos-aware scheduling of services-oriented internet of things," *IEEE Trans Ind. Informat.*, vol. 10, no. 2, 2014.

[30] I. Awan and M. Younas, "Towards qos in internet of things for delay sensitive information,"*Int. Conf. Mobile Web and Inform. Syst.*. Springer, 2013, pp. 86–94.

[31] G. Tanganelli, C. Vallati, and E. Mingozzi, "Energy-efficient qos-aware service allocation for the cloud of things," *IEEE 6th Int. Conf. Cloud Comput. Technology and Sci. (CloudCom)*. IEEE, 2014, pp. 787–792.

[32] A. S. Al Rawahi, K. Lee, J. Robinson, and A. Lotfi, "An evaluation of optimisation approaches in cloud of things resource trading," *6th Int. Conf. Future IoT and Cloud (FiCloud)*. IEEE, 2018, pp. 208–215.

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

[34] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007.

[35] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search,"*Int. Conf. Parallel Problem Solving from Nature*. Springer, 2004