An Evaluation of Optimisation Approaches in Cloud of Things Resource Trading

Ahmed Salim Al Rawahi, Kevin Lee, Jon Robinson, Ahmad Lotfi
School of Science and Technology, Nottingham Trent University, UK
ahmed.alrawahi2016@my.ntu.ac.uk; {kevin.lee, jon.robinson, ahmad.lotfi}@ntu.ac.uk

***Abstract*—Cloud Computing and Internet of Things (IoT) have evolved to meet the requirements of many real-world applications. Many of these requirements cannot be fulfilled by using either technology separately. In order to fulfil such diverse requirements, the integration of Cloud Computing and IoT is emerging as a new paradigm called Cloud of Things (CoT). CoT is expected to host heterogeneous resources and fulfil complex requirements of resource providers and consumers. This complexity poses a real challenge for resource allocation in CoT. To tackle this challenge, resource allocation is described as a trading optimisation problem and utility functions are used to rank candidate resource allocation assignments. The contributions of this paper are 1) introducing vocabularies needed for trading CoT resources 2) proposing a marketplace system architecture to enable efficient trading of CoT resources 3) examining the use of different utility functions to rank candidate resource assignments 4) performing simulations to validate the effectiveness of the proposed approach using three optimisation algorithms.**

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

I. Introduction

Cloud Computing transforms computing resources into a modern utility. Cloud resources are provided as services over the Internet rather than as physical assets. It is widely adopted in many applications such as e-learning, e-business, health, logistics and manufacturing. The physical scope of Cloud Computing is limited because it is focused on data-centres and does not interact with physical world.

IoT is a technology that can be viewed as complementary to Cloud Computing. The paradigm aims to interconnect heterogeneous things that can interact with each other and the surroundings. This can overcome the limited reachability of Cloud Computing to physical world events that are far away from Cloud data-centres. The applications of IoT spread over many domains such as logistics, transportation, defence and public safety. The main limitation of IoT resources is their constrained computational capabilities.

Due to the shortcomings of both technologies, considerable research efforts argue for a new paradigm that integrates both technologies [1]. The new paradigm is commonly known as Cloud of Things. CoT is expected to extend the limited scope of the Cloud and provide IoT with virtually unlimited Cloud resources. It enables many emerging applications that require the integration of both Cloud Computing and IoT technologies.

Latency sensitive applications (e.g. military, emergency services) benefit from the wide coverage of IoT resources in monitoring their operations while utilising the power of Cloud resources in processing and storing data. 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.

Despite the strong interest in integrating Cloud Computing and IoT, there are still many open challenges [2]. One of the major issues is how efficiently CoT resources can be shared especially the IoT physical resources. Using market-based mechanisms to commodify resources is an approach used in similar large-scale computing infrastructure to CoT such as Grids and federated Clouds [3].

The approach of trading CoT resources is motivated as follows. IoT deployments normally require considerable investment in hardware, software and maintenance. Such investment is not affordable to many communities and it slows down the rate of IoT adoption [4]. In CoT marketplace, resources are traded as commodities rather than as physical products and priced using Cloud pay-per-use pricing model. The commoditisation of CoT resources will likely to reduce the overall costs, enable sharing and reusing of IoT resources, motivate for new services and applications.

Commoditisation of CoT resources is also motivated by many technical and business benefits. Small and medium vendors are likely to invest in IoT commodities reducing the chance of monopoly and market dominance by large vendors similar to the Cloud market [5]. Competition in the emerging market is expected to improve providers' service level agreements (SLAs). It is also expected to enable hardware and software innovations when a large number of software developers and hardware makers respond to the requirements of the CoT market.

The objective of this paper is to support the integration of Cloud and IoT and to evaluate the use of optimisation approaches in trading CoT resources. The contributions of this paper are summarised as follows: 1) vocabularies needed for trading CoT resources are introduced 2) a marketplace architecture for CoT resource trading is proposed 3) the use of different utility functions to evaluate potential allocation assignments is investigated.

The remainder of this paper is as follows. Section II reviews the related work. Section III describes the proposed architecture and defines the problem of trading resources in CoT. Evaluation results are discussed in Section IV. Conclusions and future work are presented in Section V.

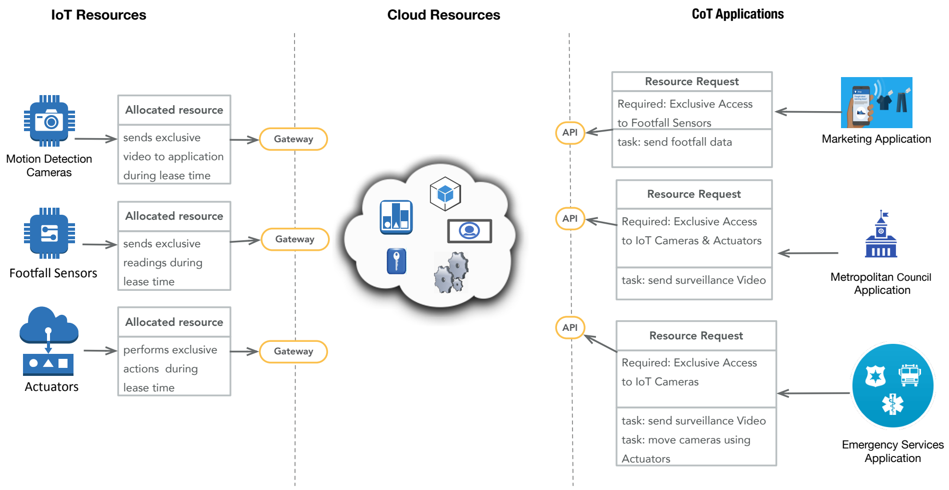
IoT Resources
Cloud Resources
CoT Applications
Motion Detection Cameras
Allocated resource
sends exclusive video to application during lease time
Gateway
Footfall Sensors
Allocated resource
sends exclusive readings during lease time
Gateway
Actuators
Allocated resource
performs exclusive actions during lease time
Gateway
API
Resource Request
Required: Exclusive Access to Footfall Sensors
task: send footfall data
Marketing Application
API
Resource Request
Required: Exclusive Access to IoT Cameras & Actuators
task: send surveillance Video
Metropolitan Council Application
API
Resource Request
Required: Exclusive Access to IoT Cameras
task: send surveillance Video
task: move cameras using Actuators
Emergency Services Application

Fig. 1. CoT applications can request exclusive utilisation of certain IoT resources. This approach enables shared access to and reusing of constrained IoT resources. It is assumed that a single exclusive access to the resource(s) from one consumer application during the lease time of the resource.

II. Related Work

Many resource management problems in large-scale computing infrastructures are non-deterministic polynomial-time hard (NP-hard) [6]. This means there are no best or exact solutions to such problems in a reasonable time due to the complexity, scalability and uncertainty of users' requirements. CoT is a large-scale computing infrastructure by nature and its resource management aspects are challenging [7], [8].

The efforts made to efficiently allocate resources when integrating Cloud Computing and IoT is described in II-A. The related work of resource allocation in CoT by commoditising resources is reviewed in II-B.

A. Resource Allocation when Integrating Cloud and IoT

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 Amazon EC2 Cloud. A device/Cloud framework has been presented in [9] 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. A scalable CoT architecture has been developed in [10] along with two algorithms to discover and virtualise IoT resources. The proposed algorithms have been developed to minimise the number of physical resources deployed and communication overhead. A detailed theoretical modelling for integrating sensors and Cloud Computing has been provided in [11] to evaluate the cost-effectiveness and performance of the architecture.

Resource allocation techniques in IoT environment are still emerging. A considerable amount of research investigates resource allocation in IoT as part of other systems (e.g. Cloud Computing, CoT, WSNs). Authors in [12] categorise approaches of resource allocation in IoT into three categories, namely, Cloud only approaches, IoT Cloud approaches and IoT only approaches. IoT Cloud approaches focus on integrating IoT resources into a Cloud as part of its services. These approaches aim to enable on-demand provisioning of shared IoT resources via the Cloud of Things.

A consensus-based framework has been developed in [13] to allocate IoT resources in the Cloud. The goal of allocation algorithm is to improve the lifetime of the connected resources. A three-tier CoT architecture has been proposed along with the development of multi-objective scheme to optimise task allocation in CoT [14]. The scheme aims to minimise the energy consumption and latency. Another three-tier architecture is designed in [15] to enable sharing of Cloud resources in vehicular networks. In this scenario, vehicles are the Things of IoT. The intent of the proposed system is to reduce service dropping rate. [16] proposes a resource allocation algorithm to enable Cloud providers optimising the throughput, occupancy and utilisation of the IoT requests.

An architecture that integrates sensors and Cloud Computing for military operations has been developed and implemented in [17]. Resource allocation in the proposed architecture is based on user prioritises to improve the performance and availability of resources for priority users. A model has been developed to cooperate between airborne sensor network and back-end Cloud in [18]. The model applies heuristics to minimise the travel time of the drones and failures in meeting their deadlines.

B. Commoditisation of CoT Resources

A solution to resource allocation problem in CoT is to enable efficient resource sharing. One of the main obstacles to this is the lack of support to share CoT resources. An emerging trend argues for market mechanisms to trade resources in large-scale infrastructures similar to CoT (e.g. Grids, Clouds, WSNs, Vehicular Networks) [3], [5], [19].

A conceptual model has been proposed in [4] to argue for the creation of trading-based value for IoT resources. The model aims to enable sharing and reusing IoT resources by trading them similarly as Cloud resources. A marketplace architecture is designed in [20] 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 [21]. 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 of the requests, minimum energy consumption of the system and maximum profit of the broker. A federation model for Cloud IoT providers has been proposed in [22] to support market mechanisms. The goal of the proposed model is to satisfy providers' requirements and improve the rate of resource utilisation of assigned tasks.

A combinatorial auction algorithm has been developed to allocate resources in CoT [23]. The objective of the algorithm is to maximise the providers' profit and rate of job completion. A reputation-based framework for CoT architectures has been presented in [24]. The framework uses auction procedure to select physical resources for sensing tasks and made payments for users. An auction model has been designed in [25] to assign CoT computation resources to the consumers. The model targets performance improvement when allocating distributed IoT resources. Another auction-based algorithm has been developed to in [26] to support resource allocation in CoT environments. The proposed algorithm aims to maximise the providers' profit while maintaining their capacity constraints.

This paper builds on [20] to evaluate the use of optimisation algorithms when trading CoT resources. The approach of using optimisation algorithms to solve this trading problem is justified due to their capabilities in finding optimal solutions to similar complex problems. 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. TRADING OF CLOUD OF THINGS RESOURCES

The rapid development of IoT hardware and software platforms reduces the costs of building and deploying IoT applications. This makes CoT deployments feasible and motivates commoditisation of its resources. Commoditised resources can be greatly utilised in high-density areas (e.g. metropolitan areas) where CoT resources can be offered to many consumers. To elaborate, the following application scenario is presented.

Various IoT nodes including sensors and actuators are deployed by multiple providers across one of the main streets in a metropolitan area. There are different Cloud providers to provide Cloud resources (e.g processing, memory and storage). Consumers can request resources for various applications. For instance, law enforcement agencies can utilise footfall and motion-detection sensors to monitor, analyse and manage public emergencies (e.g. public events management, smart evacuation planning). A metropolitan council can use environmental and footfall sensors to draw a map of activities in the area and associated pollution caused so better management and planning actions can be taken (e.g. traffic management, pedestrian pavement planning). An illustration of CoT applications is shown in Fig. 1.

A. Proposed Marketplace Architecture

For efficient commoditisation of CoT resources, global on-demand access, efficient sharing, and optimal allocation of CoT resources have to be enabled. In order to achieve this goal, a marketplace architecture for trading CoT resources is proposed in Fig. 2. The proposed architecture and the process of trading CoT resources are described in this section.

The resource request manager submits available resources/requests to the optimiser. The optimiser consists of two components, namely utility directory and optimisation tool. The utility directory maintains all utilities defined for trading proposes (e.g. cost-based utilities, time-based utilities, performance-based utilities). The optimisation tool implements most suitable optimisation techniques to efficiently optimise matching providers to consumers. The optimal assignment is submitted by the optimiser to the resource allocation manager. The scheduler maintains the resource schedule and control the lease-time of resources and manages the assignments of tasks in the Cloud. The allocator orchestrates mechanisms of joining and dis-joining resources based on scheduler plan.

This architecture is designed with consideration of flexibility and dynamism required in CoT. The optimiser is the heart of the system, requires minimal changes either by using different utility function or different optimisation technique. It reduces the time required to find a better assignment of resource allocation and increases the number of candidate solutions.

B. Problem Statement

Resource allocation in CoT is formulated as an optimisation problem where different optimisation algorithms are applied including Particle Swarm Optimisation (PSO) [27], Differential Evolution (DE) [28] and Basin Hopping (BH) [29]. These algorithms are selected for two reasons. They are gradient-free and they are well known to solve problems similar to trading CoT resources in complexity. The optimisation problem in this paper is considered as a single-objective problem and the implementation is performed accordingly.

The marketplace M consists of n number of consumers C = c_1,...,c_d who request a number of resources R = r_1,...,r_j from m number of providers P = p_1,...,p_m. Providers submit their resources and consumers submit their requests to the resource request manager of the marketplace where they have to be filtered to match the marketplace standards. The optimisation tool aims to find the optimal solution that matches resources to consumers based on a set of objectives. In CoT environment, many decision variables can be considered for optimisation. In this section, vocabularies required for trading CoT resources

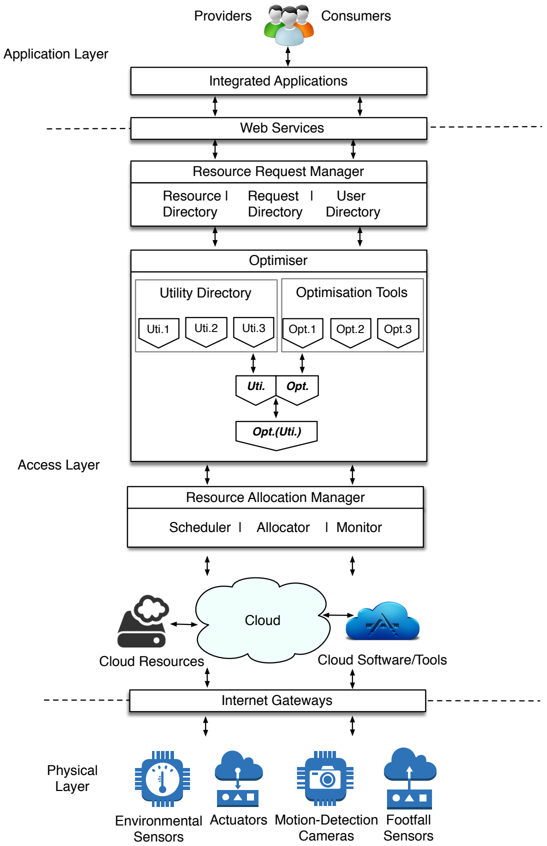

Fig. 2. CoT marketplace Architecture

are defined in table I. The objectives considered in this paper are described as follows.

Objective 1: Maximising Provider Profit. The providers always aim to maximise their profit. A utility is needed to achieve this objective. cs_j donates the cost of a resource from provider j and t_i donates the requested lease time of a resource by consumer i. The cost of allocating a resource to a consumer can be calculated as $(cs_j.t_i)$. The utility for maximising the profit of providers can be represented as follows:

$$\text{Maximise} \quad P_g = \sum_{i=1}^{n} \sum_{j=1}^{m} cs_j.t_i \tag{1}$$

The pseudocode of maximising provider profit utility is shown in Algorithm 1.

Objective 2: Maximising Resource Coverage. Consumers are expected to look for resources that provide them with the maximum area coverage when utilising resources. To achieve this goal, the sensing range of a resource s_j and the maximum transmission power level Et_{max} can be used to measure how far a resource can reach $(s_j.Et_{max})$. The requested location of resources is equal for all consumers and formulated as $A = (x_i y_i)$ that represents rectangular grids of identical dimensions. The objective of maximising

TABLE I
List of Vocabularies

vocabulary	Description
M	The marketplace
n	Number of resource consumers
m	Number of resource providers
r	Resource
R	Set of resources
c	Resource consumer
C	Set of resource consumers
p	Resource provider
P	Set of resource providers
b_i	Bid from consumer i
cs_j	Cost of a resource from provider j
CS	Total cost of set of resources
cm	commission of the marketplace
t_i	Lease time of requested resource
rq_i	Request from consumer i
RQ	set of consumer requests
E	Energy consumption
Er_i	Energy required by consumer i
Ep_j	initial power supply of a resource
Et_{max}	Maximum transmission power of resource j
l	Location of a resource
L_{ij}	Latency between consumer i and provider j
t_{start}	Time of requesting a resource from a provider j
t_{ack}	Time of receiving acknowledgement from a provider j
t_{qd}	Estimated queuing and transmitting delays
R_t	Response Time
RL_j	Requests limit of provider j
s_j	Sensing range of resource j
C_v	Area coverage of a resource
$(x_i y_i)$	Location requested by consumer i
M_g	Profit of the marketplace
P_g	Profit of a provider
cp_i	Capacity requested by consumer i
cp_j	Total Capacity of provider j

Algorithm 1 Utility for maximising provider profit

```
Input: 1) list of consumers with their requests' attributes, 2) list of
    providers with their resources' attributes.
Output: assignment(consumers, providers) of best optimised profit
Function: Provider Profit Utility Function
Initialise cost and capacity counter to zero, assignments to empty
Loop over array of initial guess
Generate an assignment of (consumers, providers)
End Loop
Loop over list of generated assignments
If length(consumers) = length(assignments) AND capacity
    Counter < provider Capacity AND consumer Bid ≤ provider
    Cost AND energy Requested ≤ energy Offered
Then, calculate Provider Profit
Else, set provider Cost to minimum value
End Loop
increase capacity Counter
calculate profit of all participated Providers in the assignment
return profit of the assignment to optimizer
END Function
```

the coverage is introduced as follows:

$$\text{Maximise} \quad C_v = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{s_j.Et_{max}}{A_i} + s_j \tag{2}$$

Algorithm 2 shows the pseudocode of a utility maximising area coverage.

Algorithm 2 Utility for maximising area coverage

```
Input: 1) list of consumers with their resource requests' attributes, 2) list of providers with their resources' attributes, 3) Area of requested resources
Output: assignment(consumers, providers) of best optimised coverage
Function: Coverage Utility Function
  Initialise capacity counter to zero, assignments to empty
  Loop over array of initial guess
  Generate an assignment of (consumers, providers)
  End Loop
  Loop over list of generated assignments
  If length(consumers) = length(assignments) AND capacity Counter $<$ provider Capacity AND consumer Bid $\leq$ provider Cost AND energy Requested $\leq$ energy Offered
  Then, calculate coverage of the requested resource
  Else, set coverage to minimum value
  End Loop
  increase capacity Counter
  calculate the coverage of all participated resources in the assignment
  return the coverage of the assignment to optimizer
END Function
```

Objective 3: Minimising Response Time. Response time is also considered one of the very important objectives to minimise in large-scale distributed systems. The latency between consumer i and provider j is donated by $L_{ij} = t_{ack} - t_{start}$ which measures the elapsed time from submitting the request by consumer i to the time of receiving an acknowledgement from a provider j. Estimated queuing and transmitting delays t_{qd} are also considered here where they can be formulated as $t_{qd} = \frac{(L_{ij})}{RQ_i}$. The objective to minimise response time R_t is proposed as follows:

$$\text{Minimise} \quad R_t = \sum_{i=1}^{n} \sum_{j=1}^{m} L_{ij} + t_{qd} \tag{3}$$

The pseudocode of the utility minimising the response time is shown in Algorithm 3.

Objective 4: Minimising Energy Consumption. Another important objective is to minimise the power consumption of matched resources while being utilised by consumers. It can be measured by the difference between the initial power supply of the resource and the estimated power consumption requested by the consumer$(Ep_j - Er_i)$. The objective of power consumption can be presented as follows:

$$\text{Minimise} \quad E = \sum_{i=1}^{n} \sum_{j=1}^{m} (Ep_j - Er_i) \tag{4}$$

Algorithm 4 presents the pseudocode of the utility minimising the energy consumption.

Algorithm 3 Utility for minimising response time

```
Input: 1) list of consumers with their resource requests' attributes, 2) list of providers with their resources' attributes
Output: assignment(consumers, providers) of best optimised response time
Function: Response Time Utility Function
  Initialise capacity counter to zero, assignments to empty
  Loop over array of initial guess
  Generate an assignment of (consumers, providers)
  End Loop
  Loop over list of generated assignments
  If length(consumers) = length(assignments) AND capacity Counter $<$ provider Capacity AND consumer Bid $\leq$ provider Cost AND energy Requested $\leq$ energy Offered
  Then, calculate response time of requested resource
  Else, set response time to maximum value
  End Loop
  increase capacity Counter
  calculate response time of all participated resources in the assignment
  return the response time of the assignment to optimizer
END Function
```

Algorithm 4 Utility for minimising energy consumption

```
Input: 1) list of consumers with their resource requests' attributes, 2) list of providers with their resources' attributes
Output: assignment(consumers, providers) of best optimised energy consumption
Function: Energy Consumption Utility Function
  Initialise capacity counter to zero, assignments to empty
  Loop over array of initial guess
  Generate an assignment of (consumers, providers)
  End Loop
  Loop over list of generated assignments
  If length(consumers) = length(assignments) AND capacity Counter $<$ provider Capacity AND consumer Bid $\leq$ provider Cost AND energy Requested $\leq$ energy Offered
  Then, calculate energy consumption of participated resource
  Else, set energy consumption to maximum value
  End Loop
  Increase capacity counter
  Calculate the total energy consumption of resources in the assignment
  return The total energy consumption of the assignment to optimizer
END Function
```

Objective 5: Maximising Marketplace Profit. In case the marketplace is non-volunteering or not a community-based, there will be fees for trading CoT resources called a marketplace commission that is donated by cm. The marketplace will aim to maximise its profit at each successful round of resource allocation. b_i is set as a bid of consumer i, cs_j donates the cost of a resource from provider j and t_i donates the requested lease time of a resource by consumer i. The commission of the market can be presented as $cm = (b_i - cs_j).t_i$. The cost of a resource is presented as cs_j. The objective to maximise the profit of the marketplace (M_g) can be formulated as follows:

$$\text{Maximise} \quad M_g = \sum_{i=1}^{n} \sum_{j=1}^{m} cm_{ij} + cs_j \tag{5}$$

The pseudocode of the utility maximising M_g is provided in Algorithm 5.

Algorithm 5 Utility for maximising marketplace profit

```
Input: 1) list of consumers with their resource requests' attributes,
    2) list of providers with their resources' attributes
Output: assignment(consumers, providers) of best optimised marketplace profit
Function: Marketplace Profit Utility Function
Initialise capacity counter to zero, assignments to empty
Loop over array of initial guess
Generate an assignment of (consumers, providers)
End Loop
Loop over list of generated assignments
If length(consumers) = length(assignments) AND capacity Counter $<$ provider Capacity AND consumer Bid $\leq$ provider Cost AND energy Requested $\leq$ energy Offered
Then, calculate marketplace profit
Else, set marketplace profit to minimum value
End Loop
Increase capacity counter
Calculate the total marketplace profit from the assignment
return The total marketplace profit to optimizer
END Function
```

Each resource provider has a limited capacity for offering its resources to consumers. The capacity of the provider has to be greater than or equal to the total capacity requested from consumers. A capacity constraint is introduced as follows:

$$\sum_{i=1}^{n} cp_i \leq cp_j \quad \text{where } j \in P \tag{6}$$

cp_i in constraint (6) donates the capacity required by consumer i while cp_j is set to total capacity of provider j.

Constraint (7) shows the cost of a resource cs_j and the bid from consumer b_i have to be positive and b_i has to be greater than or equal cs_j.

$$0 < cs_j \leq b_i \tag{7}$$

Constraint (8) ensures the initial power Ep_j of a resource and the estimated power consumption of the consumer Er_i are positive values and Er_i is less than Ep_j. The three constraints are applied together to all utility functions used in this paper.

$$0 < Er_i \leq Ep_j \tag{8}$$

IV. Evaluation

This section presents a proof of concept evaluation of trading CoT resources. A 3-tier marketplace system architecture is proposed to perform set of simulations. Simulations have the following aims: 1) evaluate the feasibility of using market mechanisms to efficiently allocate CoT resources, 2) test various utility functions to propose candidate assignments of consumers/providers or resources/requests, and 3) evaluate the use of three optimisation techniques in CoT trading setup.

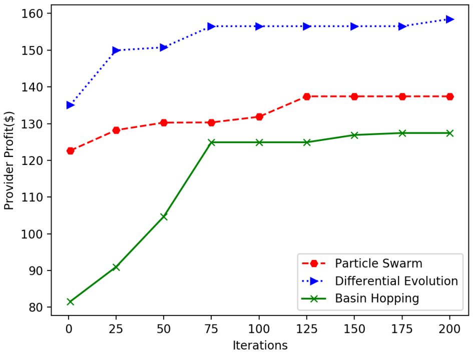

Fig. 3. Profit of Provider

A. Simulation Setup

The marketplace is assumed to find the optimal assignment of providers to consumers based on the five utilities introduced earlier. As mentioned earlier in section III-B, this paper considered the problem as a single objective optimisation problem and the utility functions are optimised individually. The scenario used in all simulations in this section is presented as follows. A number of 100 providers submit their resources to the marketplace to match them with requests of 50 consumers.

Three optimisation techniques are used to find the optimal solutions. The three techniques implemented without modification or improvement using Python programming language. A maximum number of 200 iterations is allowed for all techniques and swarm size of PSO is set to 100. Simulations are performed on a computer with the following hardware specifications: Processor: 2.6 GHz Intel Core i7, Memory: 16 GB 1600 MHz DDR3.

B. Simulation Results

This section is dedicated to discuss the results of simulations performed in this paper. Results presented in Fig. 3 to Fig. 7 compare optimal solutions found at the end of certain iterations. Table II summarises the utility values in terms of minimum, average and maximum values at the end of the last iteration.

Fig. 3 shows the provider profit utility. It is clear that DE considerably outperforms PSO and BH respectively in maximising the profit of the provider. DE and PSO maintain a steady increase in optimised profit overtime while BH experiences a sharp increase between iteration 1 and 75 before it maintains reasonable increases to the last iteration.

Fig. 4 illustrates the utility to minimise response time. It shows a competition between PSO and BH to minimise the response time while DE is clearly falling behind. BH takes more iterations than PSO to converge but both algorithms find the same optimal response time.

In Fig. 5, the utility of minimising energy consumption is illustrated. PSO is notably better than DE and BH. PSO minimises the energy consumption and converged in a fewer number of iterations than DE and BH. It is also observed

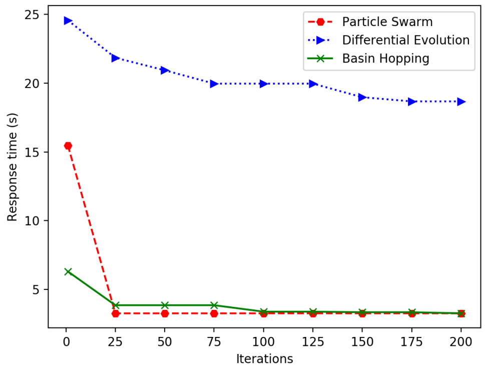

Fig. 4. Response time

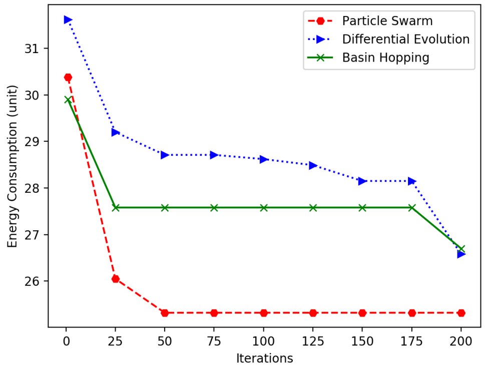

Fig. 5. Energy Consumption

all algorithms experience sharp drops between iteration 1 and iteration 25 before starting to maintain steady decreases

Fig. 6 shows the utility for maximising the coverage of requested resources. PSO outperforms the others while differential Evolution falls behind again. The three algorithms have sharp increases between iteration 1 and iteration 25 before maintaining steady increases. DE seems to be trapped by a local coverage optimal value.

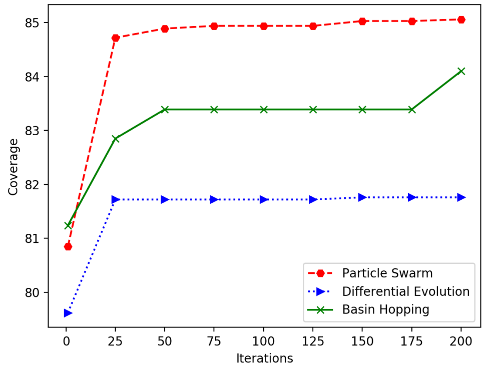

Fig. 6. Area Coverage

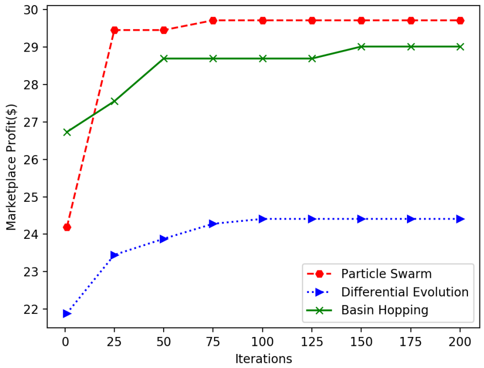

Fig. 7. Profit of marketplace

TABLE II
SIMULATION RESULTS COMPARISON

Algorithm(Utility)	Min	Avg	Max
PSO(P_g)	44.54	100.67	137.42
DE(P_g)	87.72	124.44	158.47
BH(P_g)	81.55	117.12	127.44
PSO(CV)	77.66	83.17	85.06
DE(CV)	73.20	77.55	81.76
BH(CV)	81.24	82.11	84.10
PSO(R_t)	3.27	4.08	20.20
DE(R_t)	18.67	25.86	32.42
BH(R_t)	3.27	4.37	6.49
PSO(E)	25.32	28.92	36.28
DE(E)	26.58	33.55	40.59
BH(E)	26.70	28.79	31.43
PSO(M_g)	21.30	29.18	29.71
DE(M_g)	18.30	21.63	24.41
BH(M_g)	26.41	27.36	29.01

The utility for maximising the profit of the marketplace is shown in Fig. 7. PSO and BH significantly maximise the profit of the marketplace than DE but PSO outperforms the others and converged to the optimal marketplace profit.

C. Discussion

The problem of resource allocation in CoT is presented as a single-objective trading optimisation problem. The simulation results show that the approach used in this study is promising and have several benefits. The results show the feasibility of using various optimisation algorithms as a market mechanism for trading CoT resources. Results also show at least one optimisation technique is able to find an optimum solution in all utilities proposed.

The approach taken in this paper also demonstrates that the proposed marketplace architecture can decrease the architectural complexity in CoT. The use of utility functions along with vocabularies proposed shows their effectiveness in quantifying the value of various CoT resources. This implies potential higher satisfaction for the requirements of CoT consumers/providers and higher utilisation of CoT resources.

Implementation issues are summarised as follows. 1) BH algorithm requires setting more parameters (e.g. temperature, step size, interval) than PSO and DE. It requires careful tuning of parameters to obtain better results. It is more complex than other algorithms applied and a bit slower in convergence. 2) Falling into local optima (minima and maxima) may not be avoidable in some situations by all optimisation techniques used in this paper.

V. Conclusions and Future Directions

The problem of resource allocation in CoT is investigated. Trading CoT resources using market mechanism is the approach considered to solve the problem. The problem is formulated as a single-objective optimisation problem. Vocabularies needed for trading CoT resources are introduced to provide necessary trading notations. Three gradient-free optimisation algorithms are applied to optimise utility functions including provider profit, response time, energy consumption, area coverage and profit of the marketplace. An architecture of CoT marketplace system is proposed and discussed.

Simulation results confirm the feasibility of trading heterogeneous CoT resources from multiple providers and consumed by multiple consumers. The use of utility functions along with the vocabularies proposed enabled quantifying the value of CoT resources. Results show the out-performance of PSO algorithm when compared with Differential Evolution and Basin Hopping.

Planned future work focusses on the following: 1) investigating the scalability of our approach when optimising larger sets of resources for a larger number of consumers and providers 2) Optimising more utility functions including makespan, performance and security 3) Performing further simulations with different optimisation techniques.

References

[1] K. Lee, D. Murray, D. Hughes, and W. Joosen, “Extending sensor networks into the cloud using amazon web services,” in *Networked Embedded Systems for Enterprise Applications (NESEA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1–7.

[2] 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,” *Journal of Network and Computer Applications*, vol. 67, pp. 99–117, 2016.

[3] K. Lee, G. Buss, and D. Veit, “A heuristic approach for the allocation of resources in large-scale computing infrastructures,” *Concurrency and Computation: Practice and Experience*, vol. 28, no. 5, pp. 1527–1547, 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 on*. IEEE, 2014, pp. 135–140.

[5] A. S. Alrawahi and K. Lee, “Multi-attribute combinatorial marketplaces for cloud resource trading,” in *Cloud and Green Computing (CGC), 2012 Second International Conference on*. IEEE, 2012, pp. 81–88.

[6] M. Guzek, P. Bouvry, and E.-G. Talbi, “A survey of evolutionary computation for resource management of processing in cloud computing,” *IEEE Computational Intelligence Magazine*, vol. 10, no. 2, pp. 53–67, 2015.

[7] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[8] 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 Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[9] Y. Yoon, D. Ban, S. Han, D. An, and E. Heo, “Device/cloud collaboration framework for intelligence applications,” in *Internet of Things*. Elsevier, 2016, pp. 49–60.

[10] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, “Cloud of things for sensing as a service: sensing resource discovery and virtualization,” in *Global Communications Conference (GLOBECOM), 2015 IEEE*. IEEE, 2015, pp. 1–7.

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

[12] F. C. Delicato, P. F. Pires, T. Batista *et al.*, *Resource management for Internet of Things*. Springer, 2017.

[13] V. Pilloni and L. Atzori, “Consensus-based resource allocation among objects in the internet of things,” *Annals of Telecommunications*, vol. 72, no. 7-8, pp. 415–429, 2017.

[14] 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 Computer Systems*, vol. 70, pp. 104–125, 2017.

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

[16] H. S. Narman, M. S. Hossain, M. Atiquzzaman, and H. Shen, “Scheduling internet of things applications in cloud computing,” *Annals of Telecommunications*, vol. 72, no. 1-2, pp. 79–93, 2017.

[17] 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 Systems Journal*, vol. 10, no. 2, pp. 628–636, 2016.

[18] 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 *Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 2016 Intl IEEE Conferences*. IEEE, 2016, pp. 1123–1130.

[19] S. Distefano, G. Merlino, and A. Puliafito, “A utility paradigm for iot: The sensing cloud,” *Pervasive and mobile computing*, vol. 20, pp. 127–144, 2015.

[20] A. S. Alrawahi, K. Lee, and A. Lotfi, “Trading of cloud of things resources,” in *Proceedings of the Second International Conference on Internet of Things and Cloud Computing*, ser. ICC ’17. New York, NY, USA: ACM, 2017, pp. 163:1–163:7. [Online]. Available: http://doi.acm.org/10.1145/3018896.3056780

[21] 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 of Things Journal*, vol. 4, no. 2, pp. 404–413, 2017.

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

[23] Y. Choi and Y. Lim, “Optimization approach for resource allocation on cloud computing for iot,” *International Journal of Distributed Sensor Networks*, vol. 12, no. 3, p. 3479247, 2016.

[24] B. Kantarci and H. T. Mouftah, “Trustworthy sensing for public safety in cloud-centric internet of things,” *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 360–368, 2014.

[25] S. Kim, “Nested game-based computation offloading scheme for mobile cloud iot systems,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, p. 229, 2015.

[26] 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 Internet of Things Journal*, vol. 4, no. 5, pp. 1583–1596, 2017.

[27] J. Kennedy, “Particle swarm optimization,” in *Encyclopedia of machine learning*. Springer, 2011, pp. 760–766.

[28] R. Storn and K. Price, “Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[29] D. J. Wales and J. P. Doye, “Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms,” *The Journal of Physical Chemistry A*, vol. 101, no. 28, pp. 5111–5116, 1997.