The Effects of Relative Importance of User Constraints in Cloud of Things Resource Discovery: A Case Study

Luiz H. Nunes
Federal Institute of São Paulo
Matão-SP, Brazil
lhenriquenunes@ifsp.edu.br

Júlio C. Estrella, Alexandre N. Delbem
University of São Paulo
Institute of Mathematics and Computer Science
São Carlos-SP, Brazil
{jcezar, acbd}@icmc.usp.br

Charith Perera
The Open University
Walton Hall, Milton Keynes, MK7 6AA - UK
charith.perera@ieee.org

Stephan Reiff-Marganiec
University of Leicester
University Road, Leicester, LE1 7RH - UK
srm13@le.ac.uk

ABSTRACT

Over the last few years, the number of smart objects connected to the Internet has grown exponentially in comparison to the number of services and applications. The integration between Cloud Computing and Internet of Things, named as Cloud of Things, plays a key role in managing the connected things, their data and services. One of the main challenges in Cloud of Things is the resource discovery of the smart objects and their reuse in different contexts. Most of the existent work uses some kind of multi-criteria decision analysis algorithm to perform the resource discovery, but do not evaluate the impact that the user constraints has in the final solution. In this paper, we analyse the behaviour of the SAW, TOPSIS and VIKOR multi-objective decision analyses algorithms and the impact of user constraints on them. We evaluated the quality of the proposed solutions using the *Pareto-optimality* concept.

Keywords

Internet of Things, Resource Discovery, Multi-Objective, Optimization, Sensor Search, Multiple-Criteria decision analysis

1. INTRODUCTION

Nowadays, the number of smart objects connected to the Internet is growing exponentially proportionally to the number of services and applications for them. According to the Gartner Report, there is about 6.4 billion of connected things moving a market around $235 billion just with end-users services in 2016 [13]. The integration between Cloud Computing and Internet of Things (IoT) named as Cloud

of Things (CoT) plays a key role to manage the connected things, their data and the provided services [22].

Jayaraman et al. [16] defines the Cloud of Things paradigm "*where smart objects are fully connected to the network and integrated with the cloud(s) for data storage, processing, analytics and visualization*". The number of services and applications using the CoT concepts has been increasing in several areas such as environmental monitoring, healthcare aid and assisted car driving. On the other hand, the rapidly spread in the number and type of devices makes it difficult for IoT stakeholders to use the gathered data, as generally they are used for specific purposes [28].

Solutions such as GSN, OpenIoT and Xively aims to support the CoT vision enabling access, process and analyses of smart objects and their data by using a set of keywords or semantic inference. However, due to their dynamic nature and original goal, the data of a smart object could not be suitable to accomplish the requirements of a user different of its owner.

The resource discovery process is a key challenge in the Cloud of Things context, which must to perform the smart objects search and selection regarding the constraints imposed by different users. In this sense, several research propose to use context-aware computing and multiple-criteria decision analysis (MCDA) to support the resource discovery process.

Context-aware computing refers to use stored context information to characterize a smart object and link them to their data [25]. While multiple-criteria decision analysis algorithms aims to propose the best set of smart objects according to the user objectives, constraints and their relative importance. The user constraints refer to the criteria imposed for the sensor discovery and the relative importance relate to the given weight of each criteria during the process.

Although several papers such as [22, 12, 24, 18] use some kind of MCDA to perform the resource discovery process, they are not concerned about the quality of the proposed solution set. Moreover, they do not evaluate the impact of the relative importance of user constraints in the final set of smart objects.

Thus, in this paper we present an evaluation of the impact of the user constraints and their relative importance

to select a set of smart object. In particular, we investigate the behavior of the Simple Additive Weight method (SAW), the Technique for the Order of Prioritisation by Similarity to Ideal Solution (TOPSIS) and VIseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR). To perform the experiments, the methodology presented in [23] is used to evaluate the algorithms and the impact of the user constraints in the final set.

The paper is organized as follows: Section 2 presents a literature review of resource discovery for CoT. Section 3 describes the analysed Multiple-criteria decision-making algorithms. Section 4 describes the methodology and configurations used to perform the experiments. The results are then discussed in Section 5. Finally, the conclusions and directions for future work are presented in Section 6.

2. RELATED WORK

Nowadays there are several approaches that enable the smart objects management. Perera et al. [24] and Römer et al. [27] present surveys that describes several architectures, techniques, methods, models, features, systems, applications, and middleware solutions related to the CoT context. In this section, firstly we present some architectures that enable the resource discovery of smart objects and next some works related to sensor discovery techniques.

Bovet and Hennebert [4] proposes a P2P architecture for sensor discovery aiming robustness, reliability and efficiency in energetic terms. The authors present an ontology to describe the properties, functionalities and how to access to the subscribed devices. The SPARQL language is used to look for specific devices into the ontologies, which are stored in a distributed manner over the nodes of the architecture.

Kamilaris et al. [17] use domain name server as a scalable metadata repository to support the entity discovery using their location. The authors proposes the creation of a new domain, such as *.env*, which represent the entities of the real world. Thus, when a smart object becomes available it must register their characteristics and services into the DNS repository.

Kiljander et al. [19] proposes an architecture aiming to provide smart objects interoperability. Ontologies are used to describe these devices which are accessible using SPARQL language and semantic agents. It uses unique identifiers named *ucodes* to access and identify the devices of an specific network. The *ucodes* are stored inside distributed brokers, which are organized according to their location, owners or data.

Diaz-Montes et al. [9] present the CometCloud tool to provide infrastructure and programming support to develop workflows to integrate with federated resources. CometCloud is a three-layer architecture composed by: infrastructure layer, autonomous management layer and interface layer. The infrastructure layer allows the information exchange with the distributed resources. The interface layer enables the information exchange between the user and the CometCloud core. Finally, the autonomous management layer compose the workflow according to available applications and their policies regarding the established SLA.

Carlson and Schrader [6] presents a search engine named Ambient Ocean to discovery and select sensors using context information. This search engine uses a local stored metadata to define the device context and perform the search in a more efficiently way. The search engine uses similarity multi-task

models based on the Weighted Slope One algorithm. In scenarios that is hard to model the devices features, the Ambient Ocean applies collaborative filters techniques to compute the similarity between users or sensors using previous information.

Kothari et al. [20] shows an architecture named DQS-Cloud to optimize the sensor search, autonomous fault tolerance mechanism and avoid SLA violations. The search is based on keywords and in the QoS attributes desired by the users. The DQS-Cloud aims to minimize the communication overhead reusing data flows with similar QoS levels. The results shows that the DQS-Cloud was capable to minimize the bandwidth and processing rate in the providers.

Gao et al. [12] proposes the *Automated Complex Event Implementation System* to integrate dataflows at runtime. The sensors and their flows are described according to the SSN ontology and are stored in a repository with their QoS and QoI attributes. It is able to search and select the registered sensors with regards to the specified QoS and QoI levels using the Simple-Additive-Weighting algorithm.

Perera et al. [24] present the CASSARAM framework to perform the sensor search and selection regarding user context properties. It uses the Semantic Sensor Network Ontology (SSN) to retrieve and model user context properties. CASSARAM users use semi-negotiable context properties, which allow to define context properties values in a range. Thus, the proposed Relational-Expression based Filtering can be applied to ignore irrelevant sensors during the semantic querying. Also, the Comparative-Priority Based Heuristic Filtering is used to remove the sensors that are far from the ideal point prioritizing the TOP-K selection.

Doukas and Antonelli [11] presents the COMPOSE to provide an end-to-end solution to develop applications and services for CoT. This solution operates in all layers of IoT architecture interacting with the users of the mobile application, performing the sensor search and selection and also deploy the application into the cloud. The sensor search and selection uses the iServe, which is a service warehouse that unify several features such as the service publisher, service analyse and service discovery using semantics. The iServe is able to deploy service and additional features to explore the service description, notation and analysed gathered data.

Khodadadi et al. [18] proposes a framework named Simurgh to define "things", people and their functional properties to make easier define services and compose workflows for IoT. The search and selection process uses syntax based algorithms in two phases. The first phase, look for entities that respect a specific set of criteria. On the other hand, the second phase uses the first phase result set to perform another search to choose the suitable devices for a specific problem. The framework was validate using a study case that illustrated the framework behaviour for a temperature sensor.

Nunes et al. [22] presents the ViSIoT middleware to perform the smart objects resource discovery. This work use the TOPSIS to select the sensors according to the user constraints. The ViSIoT performance analyses shows the capacity for setting up the environment in a timely manner.

The discussion presented in this section show some architectures and alternatives to perform the resource discovery and selection according to the constraints imposed by the final user. However, these works do not evaluate the quality of the proposed solutions and the effects of the user constraints in the quality of the proposed smart objects set. In

this sense, we have take the SAW and TOPSIS algorithms presented in this section as a base algorithm for a case study about the efficiency of MCDA algorithms applied in the CoT context and analyse the influence of user constraints in the final solution.

3. MULTI-OBJECTIVE OPTIMIZATION

Several problems in industry, computing, engineering and other areas uses multiple objectives optimization. In many cases, these objectives are defined in not comparable units and have some level of conflict between them. In other words, an objective can not be improved without deteriorate another objective [15]. In the sensor discovery process this scenario can be exemplified by an user which desires to choose a subset of smart objects but also wants to minimize the price and maximize the accuracy of the sensors in this subset.

In an optimization problem with one objective, the search space is always well defined. As more conflicting goals must be simultaneously optimized, it is extremely hard to establish a single optimal solution but rather a set of possibilities with equivalent quality. The optimal solution is a set of optimal trade-offs between conflicting goals [2]. An multi-objective optimization problem can be describe as

$$\text{minimize}\{f_1(x), f_2(x), \cdots, f_k(x)\}, \text{ where } x \in S, \quad (1)$$

wherein the number of objective functions k is greater than or equal to two in decision space, represented by R^n. The vector of objective functions is defined by:

$$f(x) = (f_1(x), f_2(x), \cdots, f_k(x))^T \quad (2)$$

Vector decision $x = (x_1, x_2, \cdots, x_n)^T$ belong to feasible region nonzero S, which is a subset of R^n [21].

3.1 Pareto Optimality

The Pareto dominance relationships are used to compare different sets of solutions. The set of optimal solutions of problem is given the name of set of optimal solutions or Pareto non-dominated solutions [15].

In a minimization problem, a solution x^T is not dominated if there is no $x \in S$ such that $f_i(x) \leq f_i(x^T)$ for each objective$_{i=1,\cdots,k}$ e $f_i(x) < f_i(x^T)$ for at least one of the analyzed objectives[2]. The image of the set of optimal solutions is called Pareto frontier or Pareto curve. The shape of the Pareto front indicates the nature of trade-off between different objective functions [5].

Each objective can be minimized or minimized. The solid curve represents the set of non-dominated solutions. It this figure, the optimal set of Pareto is always composed of the solutions that are concentrated in an specific vertex of a feasible region. Furthermore, in a continuous space of solutions, the optimal set of Pareto may be formed by two disjoint sets of solutions as represented by Figure(b). However, despite the existence of multiple Pareto optimal solutions in practice only one of these solutions must be used [8].

3.2 Multiple-criteria decision analysis

Multiple-criteria decision analysis algorithms are used for decision making in the presence of multiple and often conflicting goals. The MCDA algorithm are intended to assist the judgment of decision making through a set of goals and criteria, estimating their importance and establishing the contribution of each option regarding a set of criteria [10].

An MCDA problem can be described using an analysis matrix $M \times N$, where the element q_{ij} represents the performance of each option according to the decision criteria c_j in non comparable units and scales, such as represented by Equation 3. An evaluation matrix is used to represent the relative performance of each q'_{ij} using a normalization function to compare the different criteria [30].

$$Q = \begin{array}{c} \\ q_1 \\ q_2 \\ \vdots \\ q_m \end{array} \begin{array}{cccccc} c_1 & c_2 & c_3 & & c_n \\ \left[\begin{array}{ccccc} q_{11} & q_{12} & q_{13} & \cdots & q_{1n} \\ q_{21} & q_{22} & q_{23} & \cdots & q_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ q_{m1} & q_{m2} & q_{m3} & \cdots & q_{mn} \end{array} \right] \end{array} \quad (3)$$

All MCDA algorithms explicitly define its options and weights of each criterion, but differ in the way that they combine the input data. Although MCDA problems are found in different areas, they often share similar characteristics such as using multiple criteria always form a hierarchy , conflict between the criteria, hybrid nature, uncertainty and their solutions can not be conclusive [31].

3.2.1 SAW

The *Simple Additive Weight* algorithm is one of the most popular MCDA algorithms and is applied in several application domains such as supply chain management, personnel selection problems, project manager selection and facility location selection [1], [26] . The SAW algorithm aims to get a weighted sum of the normalized criterion values of each alternative, where the greater value represents the preferred alternative [26].

3.2.2 TOPSIS

The *Technique for Order of Preference by Similarity to Ideal Solution* is another popular MCDA algorithm applied in Supply Chain Management and Logistics, Design, Engineering and Manufacturing Systems, Business and Marketing Management, Health, Safety and Environment Management, Human Resources Management, Energy Management, Chemical Engineering and Water Resources Management [3]. The TOPSIS algorithm aims to choose the options that are closest to the optimal solution and farthest from the negative optimal solution [30].

3.2.3 VIKOR

The *VIseKriterijumska Optimizacija I Kompromisno Resenje* uses the concept of compromise programming and has been applied in several fields such as location selection, environmental policy and data envelopment analysis [14]. The VIKOR algorithm aims to find the options that are closest to the optimal solution, and also evaluate their individual and group impact [29].

4. EVALUATION METHODOLOGY

This Section presents the research methodology used in the experiments. We use the evaluation methodology proposed by Nunes et al. [23] to compare resource discovery algorithms from a quality of search perspective. In this Section we assume the criteria and user constraints have the same meaning and weights as well as relative importance.
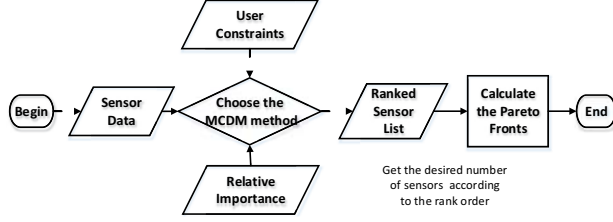
Figure 1: Evaluation Workflow. Adapted from Nunes et al. [23]

Figure 1 shows the workflow used in our experiments. The sensor data, the user constraints and their relative importance are used as input for a MCDA algorithm that will output a ranked sensor list. Next, the ranked sensor list is used as input for the Pareto Optimal Solutions Check, which define the number of optimal solutions in each Pareto front.

The metric used to evaluate the MCDA algorithms is the the Overall non-dominated vector generation ratio (ON-VGR) [7] which shows the number of optimal solutions in the Pareto front as a proportion of the number of solutions proposed by the MCDA methods in each front. The closer to one the ONVGR value is, the better is the solution proposed in that front.

The experiment environment is composed by only one physical machine, which hosts the application with multi-criteria decision algorithms. Table 1 describes the hardware used to execute the algorithms.

Table 1: Physical Environment

Hardware/Software	Specification
Processador	AMD Processor Vishera 4.2 Ghz
Memory	32 GB RAM DDR3 Corsair Vegeance
Hard Disk	HD 2TB Seagate Sata III 7200RPM
Operating System	Linux Ubuntu Server 14.04 64 Bits LTS
Java	JDK 1.7
Database	MongoDB 3.0

The experimental methodology was based on four factors: i) the number of sensors descriptions, ii) the MCDA algorithm, iii) the number of selected sensors and iv) the number of criteria. Table 2 shows the used experimental factors and levels, where the combination of the levels of each factor gives a total of 12 experiments. Each experiment was replicated one hundred times, where the criteria weights was randomly defined at execution time.

Table 2: Factors and levels used in the experiment

Factor	Level
Number of Sensors Descriptions	100,000
MCDA Method	SAW, TOPSIS and VIKOR
Number of Selected Sensors	1,000 and 10,000
Number of User Constraint	2 and 6

The sensor descriptions used as algorithm input was synthetically generated. The sensor capabilities and measurements (e.g. frequency and power consumption) are based on the 4027A Series from Bird Technologies[1]. The context data related to each sensor are retrieved from OpenWeatherMap[2]

---

[1]Bird Technologies -http://www.birdrf.com/
[2]OpenWeatherMap - http://openweathermap.org/

and their current properties values used in this experiment (e.g. battery, price, drift and response time) are assumed to be retrieved by software systems that manage such data and are available to be used.

The user constraints and objectives functions used to maximize $(\max(c_j))$ or minimize $(\min(c_j))$ follow this order: max(battery), min(price), min(drift), max(frequency), min(energy consumption), min(response time).

5. RESULTS

Figure 2 presents the boxplot representation of the ON-VGR to select 1,000 (Figure 2.a) and 10,000 (Figure 2.b) smart objects considering two user constraints. In this figure we have to suppress the outliers and limit the number of fronts to two hundred to allow the graphic view. We observe that are a high number of fronts due to the low number of user constraints conflicts, which impacts in the number of available solutions in each front.

The number of solutions available in each front increases proportionally to front index which decrease the ONVGR value because less objects are selected in these fronts. Also, the ONVGR value has a low variation when 10% of the sensors are selected rather than 1%, because more sensors are selected which increases the chances of select an optimal sensor independently of the user constraints. The mean behaviour of the algorithms shows that they could not find all the optimal smart objects, in which in the best scenario an ONVGR value closer to 0.8 and 0.9 are got when 1% and 10% of the smart objects are selected. Regarding the MCDA algorithms, we can observe the boxplot overlap each other, thus they present an equivalent behaviour when user constraints are used.

Figure 3 presents the graphic representation of the ON-VGR to select 1,000 (Figure 3.a) and 10,000 (Figure 3.b) smart objects considering six user constraints. In this Figure we observe that are less fronts than the solution presented in Figure 3 because there are more conflicts between the user constraints and consequently a high number of optimal solutions in each front. Thus, the ONVGR value is lower than the two properties scenario as the algorithms are not able to find all the solutions in the first fronts. It is important to highlight when 1% and 10% of the smart objects are selected a mean ONVGR value lower than 0.2 and around 0.4 are gotten. In this sense, the quality of the proposed solution when six user constraints are considered is worst than the proposed solution with two user constraints. About the MCDA algorithms, the three algorithms present practically the same behaviour with a slightly difference for VIKOR algorithm which uses two more fronts than SAW and VIKOR with a low number of solutions for each one.

In summary, the results show that the use of relative importance in user constraints does not necessarily improve the quality of the solution offered by a MCDA algorithm regarding the Pareto dominance relationships. The change of relative importance in user constraints has a higher impact when less constraints are used because there are less optimal solutions in each front. As expected, as more user constraints are used worst is the proposed solution. Finally, in the analysed scenarios when the relative importance of user constraints are changed there is no statistical difference between the solutions proposed by the MCDA algorithms.
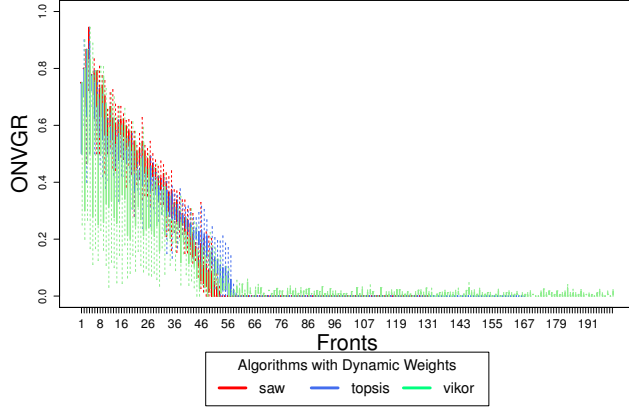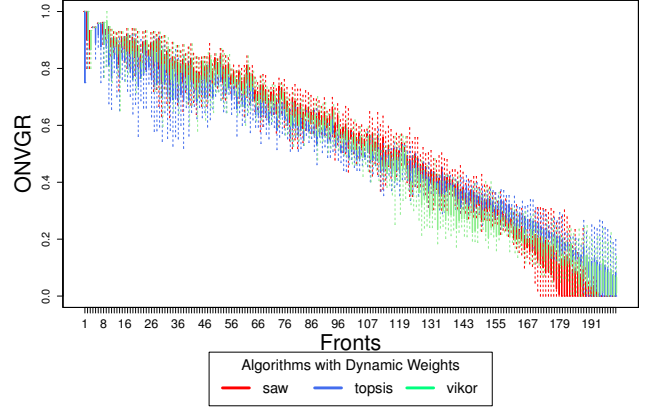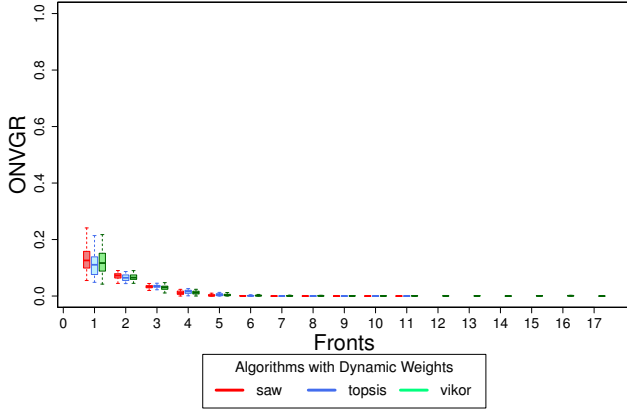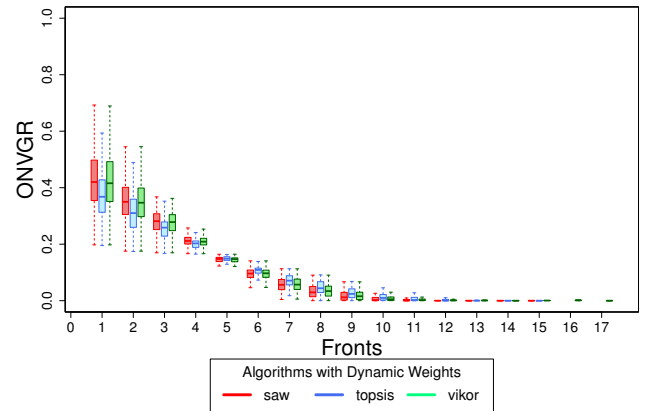
(a) Selection of 1% (b) Selection of 10%

Figure 2: Variation of ONVGR value for two user constraints

(a) Selection of 1% (b) Selection of 10%

Figure 3: Variation of ONVGR value for six user constraints

6. CONCLUSION

Efficient resource discovery of smart objects by adhering to dynamic requirements of an user is an open challenge in CoT environments. The integration of context-aware computing and multi-objective optimization has been widely used to support the sensor search and selection and to find the best trade-off between the available solution and the imposed constraints. In this paper, we have used an existent methodology presented in Nunes et al. [23] to evaluate MCDA algorithms and the impact of the relative importance of user constraints in the quality of the proposed smart object results set. The gathered results show that the use of relative importance in user constraints does not necessarily improve the quality of the solution offered by a MCDA algorithm and has a higher impact when less user constraints are used. Further, the higher number of user constraints decreases the quality of the proposed solution due to conflicts between user constraints. For future work, we will analyze other characteristics of their solutions such as convergence and distribution regarding the Pareto dominance relationships.

7. REFERENCES

[1] L. Abdullah and C. R. Adawiyah. Simple additive weighting methods of multi criteria decision making and applications: A decade review. *International Journal of Information Processing and Management*, 5(1):39, 2014.

[2] A. Abraham and L. Jain. *Evolutionary multiobjective optimization*. Springer, 2005.

[3] M. Behzadian, S. K. Otaghsara, M. Yazdani, and J. Ignatius. A state-of the-art survey of TOPSIS applications. *Expert Systems with Applications*, 39(17):13051–13069, dec 2012.

[4] G. Bovet and J. Hennebert. Distributed semantic discovery for web-of-things enabled smart buildings. In *New Technologies, Mobility and Security (NTMS), 2014 6th International Conference on*, pages 1–5, March 2014.

[5] M. Caramia and P. Dell'Olmo. *Multi-objective Management in Freight Logistics*. Springer Science Business Media, 2008.

[6] D. Carlson and A. Schrader. Ambient ocean: A web search engine for context-aware smart resource

discovery. In *Internet of Things (iThings), 2014 IEEE International Conference on, and Green Computing and Communications (GreenCom), IEEE and Cyber, Physical and Social Computing(CPSCom), IEEE*, pages 177–184, Sept 2014.

[7] Y. Collette and P. Siarry. *Multiobjective Optimization*. Springer Berlin Heidelberg, 2004.

[8] K. Deb. *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, Chichester New York, 2001.

[9] J. Diaz-Montes, M. AbdelBaky, M. Zou, and M. Parashar. Cometcloud: Enabling software-defined federations for end-to-end application workflows. *Internet Computing, IEEE*, 19(1):69–73, 2015.

[10] J. Dodgson, M. Spackman, A. Pearman, and L. Phillips. *Multi-criteria analysis: a manual*. Department for Communities and Local Government: London, 2009.

[11] C. Doukas and F. Antonelli. Developing and deploying end-to-end interoperable amp; discoverable iot applications. In *Communications (ICC), 2015 IEEE International Conference on*, pages 673–678, June 2015.

[12] F. Gao, M. I. Ali, and A. Mileo. Semantic discovery and integration of urban data streams. In *CEUR Workshop Proceedings*, volume 1280, pages 15–30, 2014.

[13] Gartner. Gartner says 6.4 billion connected "things" will be in use in 2016, up 30 percent from 2015. Avaliable in http://www.gartner.com/newsroom/id/3165317, November 2015. Last access: 23/08/2015.

[14] J.-J. Huang, G.-H. Tzeng, and H.-H. Liu. A revised VIKOR model for multiple criteria decision making - the perspective of regret theory. In *Communications in Computer and Information Science*, pages 761–768. Springer Science Business Media, 2009.

[15] A. L. Jaimes, S. Z. Martınez, and C. A. C. Coello. An introduction to multiobjective optimization techniques. 2009.

[16] P. P. Jayaraman, K. Mitra, S. Saguna, T. Shah, D. Georgakopoulos, and R. Ranjan. Orchestrating quality of service in the cloud of things ecosystem. In *2015 IEEE International Symposium on Nanoelectronic and Information Systems*, pages 185–190, Dec 2015.

[17] A. Kamilaris, K. Papakonstantinou, and A. Pitsillides. Exploring the use of dns as a search engine for the web of things. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, pages 100–105, March 2014.

[18] F. Khodadadi, A. V. Dastjerdi, and R. Buyya. Simurgh: A framework for effective discovery, programming, and integration of services exposed in IoT. In *2015 International Conference on Recent Advances in Internet of Things (RIoT)*. Institute of Electrical & Electronics Engineers (IEEE), apr 2015.

[19] J. Kiljander, A. D'Elia, F. Morandi, P. Hyttinen, J. Takalo-Mattila, A. Ylisaukko-Oja, J.-P. Soininen, and T. Cinotti. Semantic interoperability architecture for pervasive computing and internet of things. *Access, IEEE*, 2:856–873, 2014.

[20] A. Kothari, V. Boddula, L. Ramaswamy, and

N. Abolhassani. Dqs-cloud: A data quality-aware autonomic cloud for sensor services. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014 International Conference on*, pages 295–303, Oct 2014.

[21] K. Miettinen. *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media, 2012.

[22] L. H. Nunes, J. C. Estrella, L. H. V. Nakamura, R. M. D. O. Libardi, C. H. G. Ferreira, L. Jorge, C. Perera, and S. Reiff-Marganiec. A distributed sensor data search platform for internet of things environments. *International Journal of Services Computing (IJSC)*, 4(1):1–12, 2016.

[23] L. H. Nunes, J. C. Estrella, C. Perera, S. Reiff-Marganiec, and A. N. Delbem. Multi-criteria iot resource discovery: A comparative analysis. *Software: Practice and Experience*, -(-):–, 2016. *In print*.

[24] C. Perera, A. Zaslavsky, C. Liu, M. Compton, P. Christen, and D. Georgakopoulos. Sensor search techniques for sensing as a service architecture for the internet of things. *IEEE Sensors Journal*, 14(2):406–420, 2014.

[25] C. Perera, A. B. Zaslavsky, P. Christen, and D. Georgakopoulos. Context aware computing for the internet of things: A survey. *Communications Surveys & Tutorials*, vol.16:414–454, 2014.

[26] F. Ramezani, A. Memariani, and J. Lu. A dynamic fuzzy multi-criteria group decision support system for manager selection. In Y. Wang and T. Li, editors, *Practical Applications of Intelligent Systems*, volume 124 of *Advances in Intelligent and Soft Computing*, pages 265–274. Springer Berlin Heidelberg, 2012.

[27] K. Romer, B. Ostermaier, F. Mattern, M. Fahrmair, and W. Kellerer. Real-time search for real-world entities: A survey. *Proceedings of the IEEE*, 98(11):1887–1902, nov 2010.

[28] S. Sowe, T. Kimata, M. Dong, and K. Zettsu. Managing heterogeneous sensor data on a big data platform: Iot services for data-intensive science. In *Computer Software and Applications Conference Workshops (COMPSACW), 2014 IEEE 38th International*, pages 295–300, July 2014.

[29] W.-H. Tsai, W. Hsu, and W.-C. Chou. A gap analysis model for improving airport service quality. *Total Quality Management & Business Excellence*, 22(10):1025–1040, 2011.

[30] G. Tzeng and J. Huang. *Multiple Attribute Decision Making: Methods and Applications*. A Chapman & Hall book. Taylor & Francis, 2011.

[31] L. Xu and J.-B. Yang. *Introduction to multi-criteria decision making and the evidential reasoning approach*. 2001.