Multi-criteria IoT resource discovery: a comparative analysis

Luiz Henrique Nunes[1,2,*,†], Julio Cezar Estrella[1], Charith Perera[3],
Stephan Reiff-Marganiec[4] and Alexandre Cláudio Botazzo Delbem[1]

[1]*Institute of Mathematics and Computer Science (ICMC), University of São Paulo (USP), São Carlos-SP, Brazil*
[2]*Federal Institute of São Paulo (IFSP), Matão-SP, Brazil*
[3]*Faculty of Maths, Computing and Technology, The Open University, Walton Hall, Milton Keynes, MK7 6AA, UK*
[4]*University of Leicester, University Road, Leicester LE1 7RH, UK*

SUMMARY

The growth of real-world objects with embedded and globally networked sensors allows to consolidate the Internet of things paradigm and increase the number of applications in the domains of ubiquitous and context-aware computing. The merging between cloud computing and Internet of things named cloud of things will be the key to handle thousands of sensors and their data. One of the main challenges in the cloud of things is context-aware sensor search and selection. Typically, sensors require to be searched using two or more conflicting context properties. Most of the existing work uses some kind of multi-criteria decision analysis to perform the sensor search and selection, but does not show any concern for the quality of the selection presented by these methods. In this paper, we analyse the behaviour of the SAW, TOPSIS and VIKOR multi-objective decision methods and their quality of selection comparing them with the *Pareto*-optimality solutions. The gathered results allow to analyse and compare these algorithms regarding their behaviour, the number of optimal solutions and redundancy. Copyright © 2016 John Wiley & Sons, Ltd.

1. INTRODUCTION

The Internet of things (IoT) is an ecosystem that interconnects physical objects with telecommunication networks, joining the real world with the cyberspace and enabling the development of new kinds of services and applications. The IoT world is composed of small sensors and actuators embedded in the objects such as electronic devices (e.g. smartphones or tablets), clothes, alarm systems, cars, domestic appliances and industrial machines, which are capable of interacting with each other and with their environment.

Recently, the number of devices has grown rapidly, and it is anticipated that between 2015 and 2016, about 20 billion devices will be connected to the Internet creating a market of around 91.5 billion dollars [1]. These things generate an amount of data that cannot be handled in a standalone power-constrained IoT environment. The integration of IoT with cloud computing, named Cloud of things (CoT), can facilitate unprecedented ubiquitous sensing services and powerful resources to process sensing data streams beyond the capability of individual things [2].

Different domains can benefit from CoT applications such as logistics [3, 4], healthcare [5], smart cities [3], environmental monitoring [6, 7] and assisted driving [4, 8]. However, the CoT poses new challenges as it needs to combine different types of services provided by multiple stakeholders and support a large number of users and devices. One of these challenges is to provide a set of tools and

---

*Correspondence to: Luiz Henrique Nunes, University of São Paulo (USP), Institute of Mathematics and Computer Science (ICMC), São Carlos-SP, Brazil.
†E-mail: lhnunes@icmc.usp.br

environments for development of dynamic applications and ensure their seamless execution to meet the quality of context (QoC) and quality of service (QoS) requirements imposed by different kinds of applications [9].

While the nature of the CoT makes it suitable for provisioning the aforementioned services, ensuring their QoS and QoC requirements imposes complex challenges such as the resource-constrained environment, redundant data, heterogeneity of the sensors nodes, dynamic network topology and size and an unreliable communication medium. These factors can affect the user experience [10]. In addition, it is common to find two or more conflicting QoS and QoC requirements in this kind of service.

Thus, several papers such as [11–13] use some kind of multiple-criteria decision analysis (MCDA) to perform the sensor search and selection to achieve the best trade-off between the QoS and QoC properties. On the other hand, these papers do not show any concern about the quality of the selection presented by these methods regarding aspects like redundancy and dispersion of the selected sensors.

In this paper, we present a qualitative study of three MCDA methods used to establish the relative importance of multiple attributes and alternatives. In particular, we investigate the behaviour of the *simple additive weight* method (SAW), the *technique for the order of prioritization by similarity to ideal solution* (TOPSIS) and *VIseKriterijumska Optimizacija I Kompromisno Resenje* (VIKOR) under different conditions. We also analyse the quality of the solutions proposed by these methods comparing them with the available *Pareto* optimal solutions. The scientific contributions of our work can be summarized as follows:

- We proposed a methodology that can be used to compare different sensor search techniques from a quality of search perspective. We have demonstrated how our proposed methodology works using three different sensor search techniques.
- We examined the impact of 'number of context properties' towards the overall quality of sensor search.
- We examined the impact of 'number of sensors required to be selected' towards the overall quality of sensor search.
- We evaluated and compared the overall quality of sensor search between three different MCDA techniques, namely, SAW, TOPSIS and VIKOR.

The paper is organized as follows: Section 3 describes the analysed multiple-criteria decision-making algorithms and the methodology used to evaluate them. The results are then discussed in Section 4. Section 5 presents a literature review of existing approaches for sensor search and selection. Finally, the conclusions and directions for future work are presented in Section 6.

2. BACKGROUND

One of the most accepted definitions of what is IoT is described by Vermesan *et al.* [14]: 'The IoT aims to allow people and things to be connected at any time or place with anything or anyone by any path, network or service.' Usually, the IoT is considered as a three-layer architecture, as represented by Figure 1a, showing the perception layer, network layer and application layer [2, 15]. On the other hand, some authors such as Khan [16], Aazam and Huh [2] and Fersi [17] consider two extra layers named as middleware and business, as show in Figure 1b. The main layers' objectives can be summarized as follows:

- Perception layer: Its main function is to perceive and collect the real-environment information and bring them to the virtual environment. Sensors, bar code labels bar, radio-frequency identification devices, GPS and cameras are concentrated in this layer [2]. These devices can be described by metadata or specific languages such as SensorML, OGC/SWE, SSN W3C, HyperCat and semantic models to enable its use by the up layers [18].
- Network layer: It is responsible for transporting data from the perception layer to be processed. The transmission medium can use wired networks or wireless networks such as 3G, UMTS,
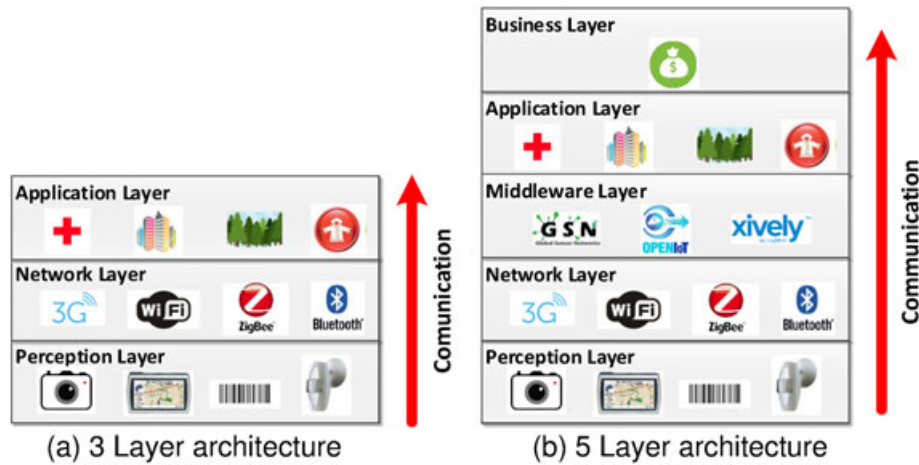
Figure 1. Inter of things architectures.

Wifi, Bluetooth, infrared and ZigBee depending directly on the types of sensor devices and the environment in which they are deployed [16].

- Middleware layer: Its goal is to offer services and store data received from the network layer. Their services must process the information and make automated decisions based on their results [2, 16]. Currently, there are several middleware solutions such as the global sensor networks (GSN) [19] and openIoT [20] to support the management of sensor networks. Usually, these solutions are able to abstract the sensors available in the perception layer and offer their resources as a service to end users.
- Application layer: It presents data from the network layer or middleware layer. This layer must be concerned to present the information according to the specifications or constraints of an user [2].
- Business layer: It is responsible for system management including its applications and services. It defines the business models, graphics and execution flows based on data received from the application layer. The success of IoT depends directly on establishing good business models to analyse the results and determine future business strategies [16].

Nowadays, there is much research conducted for the different layers of the IoT architecture, which aim to solve problems related to interoperability, scalability, reliability, data management, privacy and security. One of the most significant challenges involves the middleware layer. Specifically, concentrating on how to support the search and selection of sensors regarding the QoS and QoC properties determined by a user [9].

3. MULTIPLE-CRITERIA DECISION ANALYSIS

Multiple-criteria decision analysis refers to making decisions in the presence of multiple, usually conflicting, criteria [21]. MCDA algorithms aims to aid in the judgement of the decision-making team using a set of objectives and criteria, estimating their relative importance weights and establishing the contribution of each option regarding to each performance criterion [22].

An MCDA problem can be described using an analysis matrix ($M \times N$) in which element q_{ij} represents the performance of the option according to the decision criterion c_j in different and non-comparable units and scales, as represented in Equation (1). The evaluation matrix is used to represent the relative performance of q'_{ij} using a value/utility function to enable comparisons between the different criteria [23].

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

All MCDA algorithms explicitly define their options and contributions to each criterion, but differs in how they combine the input data. Although MCDA problems are found in different contexts, they usually share common features such as multiple attributes/criteria often forming a hierarchy, conflict among criteria, hybrid nature, uncertainty, large scale and assessments that may not be conclusive [21]. Sections 3.1– 3.3 describe three MCDA methods, namely, SAW, TOPSIS and VIKOR algorithms for MCDA. Section 3.4 proposes the use of the *Pareto*-optimality based on the proposed criterion to evaluate the applied MCDAs to select a subset of sensors.

3.1. Simple additive weight

The SAW method is one of the most popular MCDA methods [24, 25]. It provides the additive properties to calculate the final score of alternatives used for weight determinations and preferences, which is the basis of other MCDA methods such as the Analytic Hierarchy Process and Preference ranking organization method for enrichment evaluation [24]. According to [25], SAW is used in several application domains such as supply chain management, personnel selection problems, project manager selection and facility location selection.

Simple additive weighting uses an evaluation score to rank each available option. The score is obtained using a normalized criteria value multiplied by a weight. The options are ranked in descending order according to their final score, which is the sum of the scores for individual criteria [26]. SAW algorithm can be summarized by the following three steps [23]:

1. Normalize the analysis matrix **Q** described in Equation (1) to **Q'** according to Equation (2) if the criterion should be maximized or the Equation (3) if the criterion should be minimized.

$$q'_{ij} = \frac{q_{ij} - q_j^{\min}}{q_j^{\max} - q_j^{\min}} \qquad \text{for a criterion to be maximized} \qquad (2)$$

$$q'_{ij} = \frac{q_j^{\max} - q_{ij}}{q_j^{\max} - q_j^{\min}} \qquad \text{for a criterion to be minimized} \qquad (3)$$

2. Compute the score vector ϕ of each available option. Each score q'_i can be calculated using Equation (4), where w_j corresponds to the criterion weight and N represents the number of criteria in the evaluation matrix.

$$\phi(q_i) = \sum_{j=1}^{N} w_j \cdot q'_{ij} \qquad (4)$$

3. Sort options q_i in decreasing order according to the score $\phi\ (q'_i)$ to obtain the ranking of suitable options.

3.2. Technique for the order of prioritization by similarity to ideal solution

Technique for the order of prioritization by similarity to ideal solution explores the attribute information to provide a set of ranked alternatives and requires independent attribute preferences. The application domains that uses the TOPSIS method have been 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 [27]. TOPSIS sorts a set of options

according to the Euclidean distance from the ideal and negative-ideal solutions. Each option is normalized using a specific criterion value. The ideal solution represents the most desirable level of each criterion across the options under consideration, while the negative-ideal solution reflects the worst-desirable level of each criterion. The options are ranked regarding their closeness to the ideal solution and farness to the negative-ideal solution [23]. The TOPSIS algorithm can be summarized in the following steps [28]:

1. Normalize the analysis matrix **Q** to **Q'** according to Equation (5):

$$q'_{ij} = \frac{q_{ij}}{\sqrt{\sum_{i=1}^{N}(q_{ij})^2}} \tag{5}$$

   where N represents the number of options in the evaluation matrix.

2. Determine the positive ideal points (p_{+j}) and the negative ideal points (p_{-j}) of all objective functions using the analysis matrix. For a maximization criterion, the positive ideal and the negative ideal points can be calculated using Equations (6) and (7), respectively:

$$p_{+j} = \max_i(q'_{ij}) \tag{6}$$

$$p_{-j} = \min_i(q'_{ij}) \tag{7}$$

3. Compute the distances to the positive ideal solution and (s_{i+}) and the negative ideal solution (s_{i-}). The distance of each option q' to the ideal solution p_{+j} and the ideal negative solution p_{-j} is given by Equations (8) and (9):

$$s_{i+} = \sqrt{\sum_{j=1}^{n}(q'_{ij} - p_{+j})^2} \quad \text{and} \tag{8}$$

$$s_{i-} = \sqrt{\sum_{j=1}^{n}(q'_{ij} - p_{-j})^2} \tag{9}$$

4. Calculate the relative closeness to the ideal solution. The relative closeness of **q** to p_{+j} and p_{+j} represented by (c_{i+}) can be calculated according to Equation (10).

$$c_{i+} = \frac{s_{i-}}{s_{i+} - si-} \tag{10}$$

5. Sort options q_i in increasing order according to the relative closeness to c_{i+}.

3.3. VIseKriterijumska Optimizacija I Kompromisno Resenje

The basic concepts of VIKOR are a compromise programming used to obtain the most satisfactory option by the results of the individual and group regrets. This method has been widely used in several applications fields, such as location selection, environmental policy and data envelopment analysis [29].

VIseKriterijumska Optimizacija I Kompromisno Resenje introduces the multicriteria ranking index based on the particular measure of closeness to the ideal solution. The alternatives are evaluated according to all established criteria and ranks them according to (i) the minimal distance to the ideal point, (ii) the maximum group utility for the majority, and (iii) the minimum individual regret of the opponent. VIKOR algorithm can be summarized according to the follow steps [23], [28]:

1. Determine the best and the worst values for all criteria in **Q**. For a maximization criterion, the best and worst criteria values represented by q_j^* and q_j^- can be calculated respectively according to Equations 11 and 12:

$$q_j^* = \max_i(q'_{ij}) \tag{11}$$

$$q_j^- = \min_i(q'_{ij}) \tag{12}$$

2. Compute the utility measure and the regret measure. The utility measure represented by S_i is used to show the average gap of our options and can be calculated according to Equation (13), where w_j corresponds to the criteria weights, expressing their relative importance. A regret measure represented by R_i is used to show the maximal gap for improvement priority and it can be calculated according to Equation (14).

$$S_i = \sum_{j=1}^{n} \frac{w_j |q_j^* - q_{ij}|}{|q_j^* - q_j^-|} \tag{13}$$

$$R_i = \max_j \frac{w_j |q_j^* - q_{ij}|}{|q_j^* - q_j^-|} \tag{14}$$

3. Compute the group utility represented by Q_i of each solution. The v parameter is used to represent the weight of the strategy of 'the majority of criteria'. Equation (15) is used to calculate Q_i.

$$Q_i = \frac{v \cdot (S_i - S^*)}{(S^- - S^*)} + \frac{(1 - v) \cdot (R_i - R^*)}{R^- - R^*}, \tag{15}$$

where

$$S^* = \min_j S_i \text{ and } S^- = \max_j S_i$$
$$R^* = \min_j R_i \text{ and } R^- = \max_j R_i$$
$$v = 0.5$$

4. Sort options q_i in decreasing order according to the values S_i, R_i and Q_i. The results are three ranking lists.
5. Propose as a compromise solution the alternative q, which is ranked the best by the measure Q(minimum) if the following two conditions are satisfied:

    C1. Acceptable advantage

    $$q_{i+1} - q_i \geqslant DQ,$$

    where $DQ = \frac{1}{(N-1)}$ and N is the number of options
    C2. Acceptable stability in decision-making
    The alternative q_i must also be the best ranked by S or/and R.
    If one of the conditions is not satisfied, then a set of compromise solutions is proposed, which consists of the following:

    - Alternative q_i and q_{i+1} if only condition C2 is not satisfied, or
    - Alternative $q_i, q_{i+1}, \ldots, q_n$ if condition C1 is not satisfied; and q^n is determined by the relation $q_n - q_i < DQ$ for maximum n.

In summary, the methods presented in this section are used to compute the relative importance of multiple criteria and solutions based on a weighting strategy. They have been successfully applied to several real-world scenarios, where multiple conflicting objectives should be satisfied. The next section describes how the quality of the selection of sensors provided by each algorithm can be evaluated. We also present all performed experiments and the environment where they were executed.

3.4. Proposal of evaluation of multiple-criteria decision analysis methods

This section presents the research methodology used in the experiments. As a base of our study, we assume the SAW algorithm used by Gao *et al.* [13] and compare it with other popular MCDA (TOPSIS and VIKOR) algorithms. Our evaluation approach is based on a set of sensor data that will be ranked according to an MCDA method and context properties. The desired number of sensors are retrieved from the top of the ranked list, and the *Pareto*-optimal fronts are calculated. Figure 2 synthesizes the whole processing proposal for evaluating MCDAs.

The *Pareto*-optimality criterion [30] is used to compare the quality of the solutions obtained by each method. It uses the dominance concept to determine when a solution is better than other. For example, given two solutions x and y, x dominates y ($x \succeq y$) if two conditions are respected:

1. The x solution is better than y in at least one objective function;
2. The x solution is at least equal to y in all objective functions;

The set of non-dominated solutions is named *Pareto*-optimal set, which represents the set of optimal available solutions for the problem. The *Pareto* fronts is the set of values of the objective functions of the *Pareto*-optimal solutions set. The solutions that are dominated only for the *Pareto*-optimal solutions are located in the second *Pareto* front. The number of *Pareto* fronts that are used in an experiment are directly proportional to the number of non-dominated solution. In this sense, our evaluation process will consider the number of used sensors in the *Pareto*-optimal set and the number of *Pareto* fronts used by each MCDA solution. The *Pareto* fronts are computed through the *fast-non-dominated-sort* algorithm described by Deb *et al.* [31].

We considered two metrics to evaluate the MCDA methods: (i) the number of fronts, which indicates the MCDA method with more non-dominated solutions; and (ii) the overall non-dominated vector generation ratio (ONVGR) [32] metric, 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. As closer the ONVGR value is to one, the better is the solution proposed in that front.

The test environment is composed by one physical machine. Table I describes the hardware and software specification used to perform the experiments.

The experimental methodology was based on four factors: (i) the number of sensors descriptions, (ii) the MCDA method, (iii) the number of selected sensors, and (iv) the number of context properties required. In this context, the term context properties will be used to refer to the analysed sensor criteria. Table II shows the used experimental factors and levels, where the combination of the levels of each factor gives a total of 45 experiments.
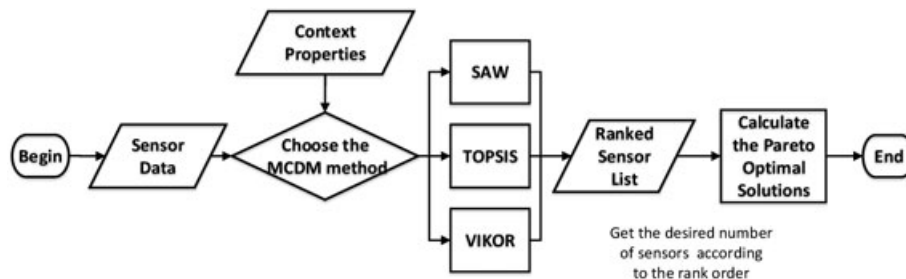

Figure 2. Evaluation workflow. MCDM, multiple-criteria decision-making; SAW, simple additive weight; TOPSIS, technique for the order of prioritization by similarity to ideal solution; VIKOR, VIseKriterijumska Optimizacija I Kompromisno Resenje.

Table I. Physical environment.

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

Table II. 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	1000 , 5000 and 10 000
Number of context properties	2,3,4,5 and 6

MCDA, multiple-criteria decision analysis; SAW, simple additive weight; TOPSIS, technique for the order of prioritization by similarity to ideal solution; VIKTOR, VIseKriterijumska Optimizacija I Kompromisno Resenje.

We assume that sensor descriptions such as sensor capabilities and measurements (e.g. frequency and power consumption) are based on the 4027A Series from Bird Technologies[†]. Similarly, we assume that context data related to each sensor are retrieved from OpenWeatherMap[‡], and their current property 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 criteria and objectives functions used to maximize ($\max(c_j)$) or minimize ($\min(c_j)$) the criteria follow this order: max(battery), min(price), min(drift), max(frequency), min(energy consumption), min(response time).

4. EVALUATION RESULTS AND LESSONS LEARNED

In this section, we present the gathered data of the performed experiments. In order to make the data visualization and their meaning easier, we will present the results of each method regarding the number of context properties.

4.1. Evaluation results

We will analyse the results regarding the SAW, TOPSIS and VIKOR methods. Two graphics represent the number of used fronts and the ONGVR metric. To represent the number of used front, the graphic have two ordinate axis. The abscissa axis has the indexes of the *Pareto* fronts from the first front to the last one . The left ordinate axis presents the number of solutions retrieved by each method (different colours lines) from each *Pareto* front. The right ordinate axis corresponds to number of *Pareto* fronts needed to cover a given subset of sensors. To represent the ONVGR metric, a graphic with one ordinate axis and one abscissa axis is used. The ordinate axis corresponds to the ONVGR value, and the abscissa axis has the indexes of the *Pareto* fronts.

4.1.1. Selection using six context properties.
Figure 3 presents the quality behaviour of the selection of 1000 (Figure 3a), 5000 (Figure 3b) and 10 000 (Figure 3c) of available sensors considering six context properties (as defined in Section 3.4). The number of *Pareto* front slightly increases as the

---

[†]Bird Technologies – http://www.birdrf.com/
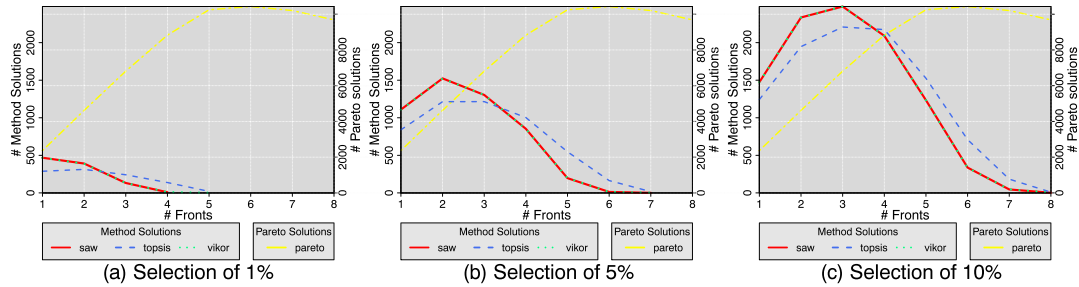[‡]OpenWeatherMap – http://openweathermap.org/

Figure 3. *Pareto* fronts for six context properties. SAW, simple additive weight; TOPSIS, technique for the order of prioritization by similarity to ideal solution; VIKOR, VIseKriterijumska Optimizacija I Kompromisno Resenje.
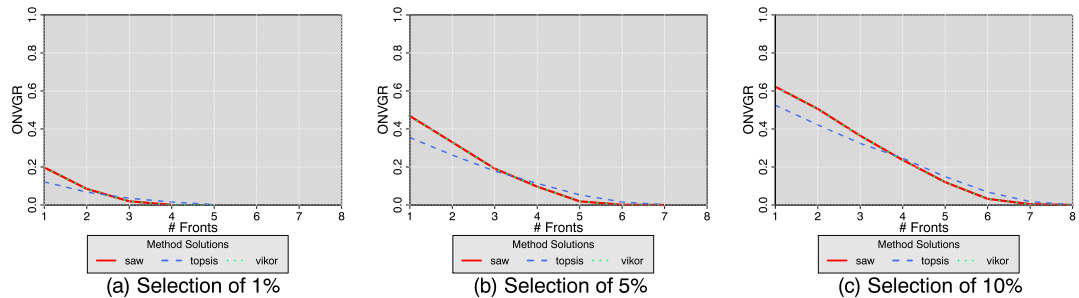

Figure 4. Overall non-dominated vector generation ratio (ONVGR) metric for six context properties. SAW, simple additive weight; TOPSIS, technique for the order of prioritization by similarity to ideal solution; VIKOR, VIseKriterijumska Optimizacija I Kompromisno Resenje.

number of selected sensors is raised. Also, the number of optimal sensors available in each front increases according to the number of selected sensors. The MCDA methods concentrates the major part of the solutions in the first fronts because of a high number of conflicts between the used criteria.

Figure 4 shows the ONVGR value in each front of the selection of 1000 (Figure 4a), 5000 (Figure 4b) and 10 000 (Figure 4c). The ratio value in the first fronts increases proportional to the number of selected sensors. On the other hand, the ratio value shows a high loss of optimal sensors as the ratio values changes from 0.2 to 0.6 in the worst and best scenarios, respectively.

The MCDA methods does not use the *Pareto* optimality concept to select the sensors. They aim to select sensors that present a certain level of stability between the context properties values, while the *Pareto* optimality solutions do not care about the stability between the context properties values but try to obtain the greatest number of context properties with the best possible values.

Regarding the MCDA methods, the TOPSIS method presented the worst solution as it shows the lowest ratio value of the analysed MCDA methods in all scenarios. In addition, the SAW method is slightly better than VIKOR method when 1% of the sensors were desired as it uses less fronts, while the proposed solutions when 5% and 10% of the selected sensors were equivalent.

4.1.2. Selection using four and five context properties. Figures 5 and 6 present the quality behaviour of the selection of 1000 (Figures 5a and 6a), 5000 (Figures 5b and 6b) and 10 000 (Figure 5c and 6c) of available sensors considering four and five context properties, respectively . Analogous to Section 4.1.1, the number of *Pareto* front and the number of optimal solutions increases proportional to the number of selected sensors. For four and five context properties, the number of *Pareto* fronts is twice as the results presented in Section 4.1.1 and are not so different; it varies from 6 to 16.

Figures 7 and 8 show the ONVGR value in each front of the selection of 1000 (Figures 7a and 8a), 5000 (Figures 7b and 8b) and 10 000 (Figures 7c and 8c). Although the number of solutions found by each method and the number of *Pareto* solutions are different from Section 4.1.1, the ratio between the number of selected sensors and the number of *Pareto* solutions are practically the same as pre-
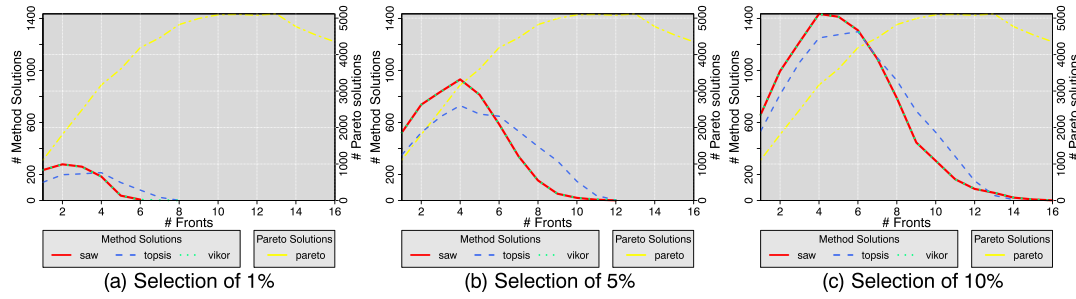
Figure 5. *Pareto* fronts for four context properties. SAW, simple additive weight; TOPSIS, technique for the order of prioritization by similarity to ideal solution; VIKOR, VIseKriterijumska Optimizacija I Kompromisno Resenje.

Figure 6. *Pareto* fronts for five context properties. SAW, simple additive weight; TOPSIS, technique for the order of prioritization by similarity to ideal solution; VIKOR, VIseKriterijumska Optimizacija I Kompromisno Resenje.

Figure 7. Overall non-dominated vector generation ratio (ONVGR) metric for four context properties. SAW, simple additive weight; TOPSIS, technique for the order of prioritization by similarity to ideal solution; VIKOR, VIseKriterijumska Optimizacija I Kompromisno Resenje.
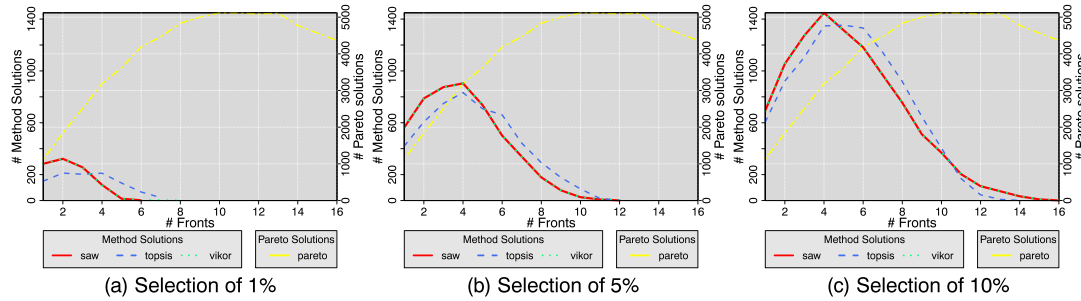

Figure 8. Overall non-dominated vector generation ratio (ONVGR) metric for five context properties. SAW, simple additive weight; TOPSIS, technique for the order of prioritization by similarity to ideal solution; VIKOR, VIseKriterijumska Optimizacija I Kompromisno Resenje.
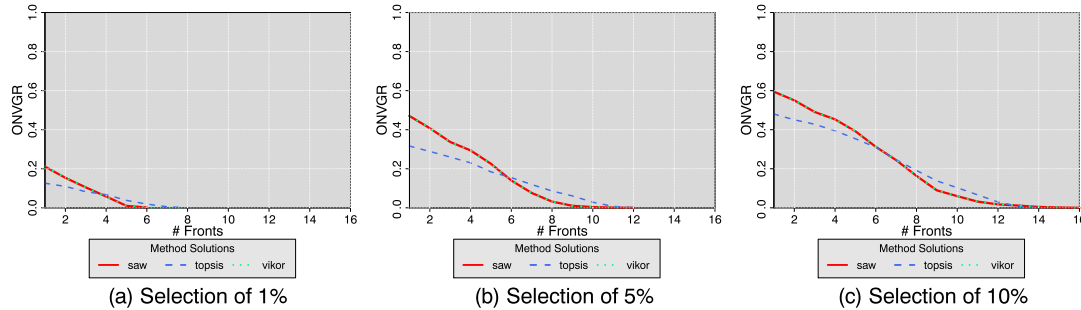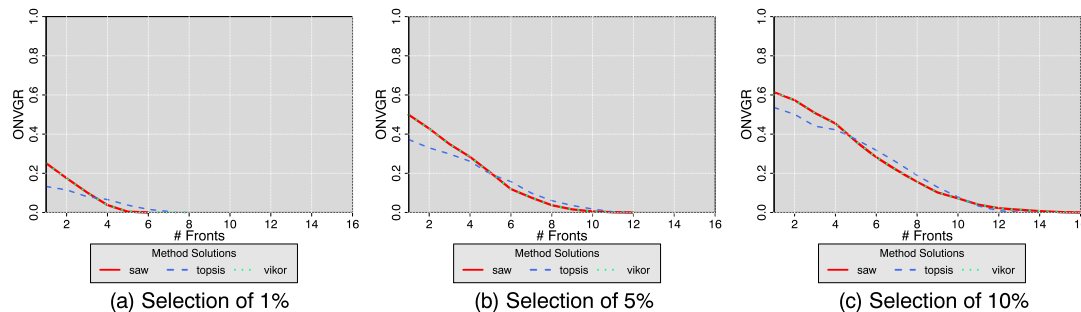
sented in Figure 4. They also show a low ratio value that changes approximately from 0.2 to 0.6 in the best and worst scenarios, respectively.

Considering the MCDA methods, the solution proposed by the SAW method is again slightly better than the solution proposed by the VIKOR method when 1% of the sensors were desired; while the proposed solutions when 5% and 10% of the selected sensors were equivalent. On the other hand, TOPSIS presents a lower quality solutions as it shows a minor number of sensors in the top first fronts.

4.1.3. Selection using three context properties. Figure 9 presents the quality behaviour of the selection of 1000 (Figure 9a), 5000(Figure 9b) and 10 000 (Figure 9c) of available sensors considering three context properties. As seen in Section 4.1.2, the number of *Pareto* front increases proportional to the number of selected sensors. This observation is justified because with less context properties, we also reduce the number of context properties conflicts, the number of *Pareto* optimal solutions per front and the number of solutions found per front, which increases the probability for finding solutions with a higher level of stability.

Figure 10 shows the ONVGR value in each front of the selection of 1000 (Figure 10a), 5000(Figure 10) and 10 000 (Figure 10c). The ratio value in the first fronts are slightly higher than the ratio values presented in Section 4.1.2 because of the reduction of the number of criteria. In this sense, the ratio values changes from approximately 0.4 to 0.8 in the worst and best scenarios, respectively.

Moreover, when the MCDA methods are analysed, the quality of the solution proposed by the SAW method was slighter better than the quality of the solution proposed by VIKOR method when 1%, 5% and 10% of the available sensors were selected as the SAW solution uses less *Pareto* fronts. Similar to Section 4.1.2, the TOPSIS method presented the solution with low quality as it had less solutions than the SAW and VIKOR methods in the top first fronts.

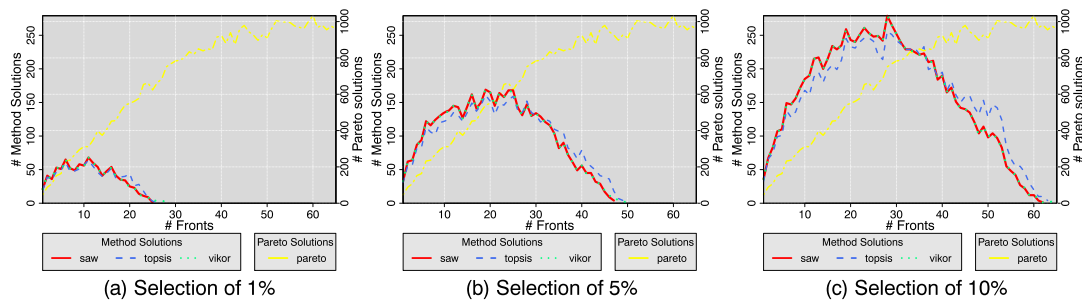(a) Selection of 1%   (b) Selection of 5%   (c) Selection of 10%

Figure 9. *Pareto* fronts for three context properties. SAW, simple additive weight; TOPSIS, technique for the order of prioritization by similarity to ideal solution; VIKOR, VIseKriterijumska Optimizacija I Kompromisno Resenje.

(a) Selection of 1%   (b) Selection of 5%   (c) Selection of 10%
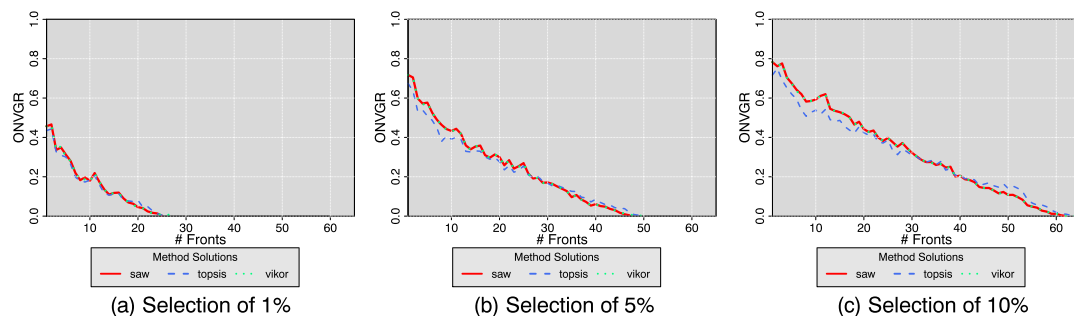
Figure 10. Overall non-dominated vector generation ratio (ONVGR) metric for three context properties. SAW, simple additive weight; TOPSIS, technique for the order of prioritization by similarity to ideal solution; VIKOR, VIseKriterijumska Optimizacija I Kompromisno Resenje.

4.1.4. Selection using two context properties. Figure 11 presents the quality behaviour of the selection of 1000 (Figure 11a), 5000(Figure 11b) and 10 000 (Figure 11c) of available sensors considering two context properties. Analogous to Section 9, the number of *Pareto* fronts and the number of optimal solutions increases directly proportional to the number of selected sensors.

Figure 12 shows the ONVGR value in each front of the selection of 1000 (Figure 12a), 5000 (Figure 12b) and 10 000 (Figure 12c). Because of the minimal number of criteria conflicts, the ratio value tends to get closer to one as its less optimal solutions are available. In this sense, the ratio changes from approximately 0.8 to 1 in the worst and best scenarios, respectively, which shows that all optimal solutions are selected.

Furthermore, the SAW method presented again the solution with better quality independently of the selected sensors numbers as it presented a high ONVGR value and uses less fronts than VIKOR and TOPSIS. The solution presented by VIKOR was quite similar to the solution presented by the SAW method, but its solution uses more fronts than SAW. The TOPSIS method presented the poorest solution, as it shows a higher number of fronts and a low ONGVG value in the top first fronts.

4.2. Lessons learned

In this section, we have compared the behaviour and quality of different MCDA methods for sensor search and selection. Firstly, it is important to highlight the number of optimal solutions available in each scenario. As expected, the number of optimal solutions increases proportional to the number of fronts. It occurs due to the non-dominated solution concept used to compute the optimal solutions set in each front. In this sense, the number of optimal solutions is not influenced by the number of selected sensors. On the other hand, the number of selected sensor affects the number of optimal solutions that are founded by the MCDA algorithms. The influence of the number of selected sensors can be justified, because it increases the chances of the MCDA find the optimal sensors set. In all

(a) Selection of 1%   (b) Selection of 5%   (c) Selection of 10%

Figure 11. *Pareto* fronts for two context properties. SAW, simple additive weight; TOPSIS, technique for the order of prioritization by similarity to ideal solution; VIKOR, VIseKriterijumska Optimizacija I Kompromisno Resenje.

(a) Selection of 1%   (b) Selection of 5%   (c) Selection of 10%

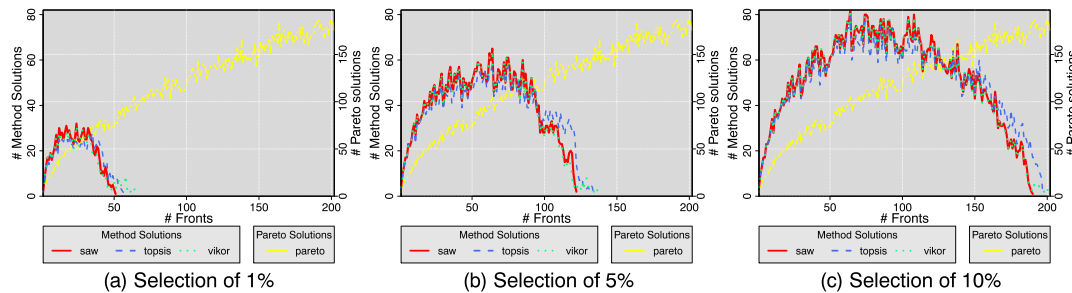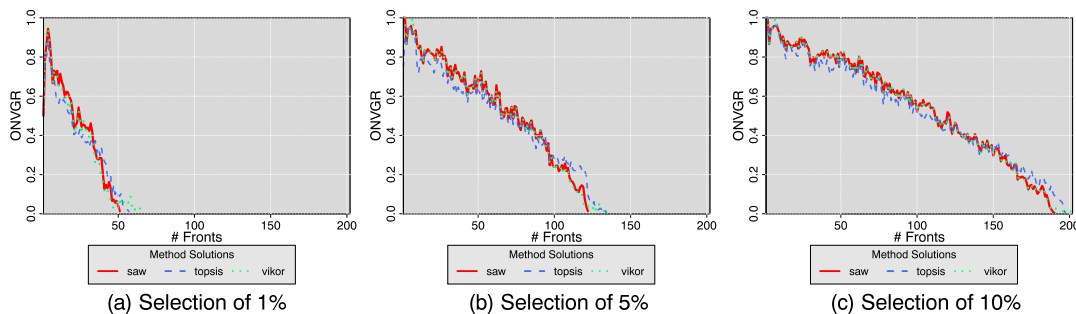Figure 12. Overall non-dominated vector generation ratio (ONVGR) metric for two context properties. SAW, simple additive weight; TOPSIS, technique for the order of prioritization by similarity to ideal solution; VIKOR, VIseKriterijumska Optimizacija I Kompromisno Resenje.

scenarios, the ONVGR metric clearly shows the significant increase of the ratio between the number of optimal sensors and the number of sensors found by each MCDA algorithm when more sensors are selected.

The context properties also influence the number of optimal solutions obtained by each MCDA algorithm. The number of context properties is directly proportional to number of optimal solutions available in each front. It is because as more context properties are used, the number of conflicts between the criteria increases and consequently the number of non-dominated solutions increases. In other words, we reduce the chances to find a small set of solutions that present the best trade-off between the analysed context properties.

Also, the ONVGR metric allows to compare how the number of context properties influence the number of optimal selected sensors. Although the number of selected sensors in each front is different for six, five or four context properties, the ONVGR value is practically the same for all and indicates that a low number of optimal sensors is founded in each one. When three or two context properties are used, the ONVGR value is higher for all scenarios, and consequently, a higher number of optimal sensors is founded when less context properties is used.

Regarding the analysed MCDA methods, it is possible to observe that for all analysed scenarios, the SAW method, which uses regular arithmetical operations of multiplication and addition to rank the options, presented at least an equal number of fronts and the ONVGR value than TOPSIS and VIKOR method. The VIKOR method, which apply the compromise programming concept providing a maximum group utility for the majority and a minimum of an individual regret for the opponent, presented a solution pretty closer to the proposed solution by SAW algorithm, but in some scenarios, its solution has more fronts. Finally, the TOPSIS method, which ranks the solutions according to the distance to the ideal solution and the greatest distance from the negative-ideal solution without consider the relative importance of these distances, presented the poorest solution as in the major part of the scenarios, it uses more fronts and presented a low ONVGR value than SAW and VIKOR methods.

5. RELATED WORK

Today, there are several approaches that enable the sensor management. Perera *et al.* [33] and Römer *et al.* [34] present surveys that describes several techniques, methods, models, features, systems, applications and middleware solutions related to the IoT context. These surveys shows that the algorithms used to perform the sensor search and selection can be splitted in two groups: prediction models and keyword or context information. In this section, we present the main work related to each group.

Elahi *et al.* [35] presents a primitive called *sensor ranking* to perform the sensor search in an efficient way. The main idea of sensor ranking primitive is to explore the periodicity presented by the sensor in some cases using prediction models that rank the sensors according to the probability to meet a user query. The single-period and the multi-period predictions models are used in this paper, and the gathered data allow to observe a performance improvement to select the sensors. Ostermaier *et al.* [36] present a search engine for the Web of Things called *Dyser* to conduct searches in scalable environments with highly dynamic content. *Dyser* is able to collect and store data and information from sensors to allow search based on metadata. It also extends the work presented by Elahi *et al.* [35] using the aggregated prediction model. The results showed that the algorithms presented a better quality selection when compared with the random model.

Truong *et al.* [37] also extends the work presented by Elahi *et al.* [35] and propose a prediction model based on fuzzy logic named time-independent prediction model. This model is able to detect anomalies about sensor behavior using metrics of density and stability. The density metric is used to estimate the probability of a certain value belonging to a specific sensor, while the stability metric estimates the stability of these sensors in the past. The combination of these metrics allows to rank the sensors and check their state. Thus, the solution presented is able to reduce the necessary communication for sensor search and selection.

Carlson and Schrader [38] present a search engine named Ambient Ocean to search and select sensors using context information. The search engine uses metadata, which is stored in a global

repository, to establish the sensors context and carry out the search in a more efficient and effective manner. Ambient Ocean uses multi-task similarity models based on the weighted slope one algorithm to select the sensors. In scenarios where the characteristics of the sensors are difficult to model, collaborative filtering techniques are employed to compute similarities between users or sensors based on information history.

Ding *et al.* [39] propose a hybrid search engine to IoT environments, able to perform searches using quantitative values, keywords and spatio-temporal relations. The architecture of this search engine is based on a bottom-up model with three layers, the first layer is responsible for sensing and monitor the equipment. The second layer is responsible to store the data in a distributed form. The third layer provides optimized access to data from the sensors. The search for keywords and quantitative values is optimized by a B+ tree and the search base on time-space relationships uses an R tree. This search engine allows the discovery of the objects state at run-time as the sensors sends continuous data to the storage layer, which index these data according to data structure used.

Guinard *et al.* [40] propose a module for the integration architecture named SOCRADES, which aims to enable ubiquitous integration services running on embedded with other business processes devices. The proposed module is based on the model Publish/Subscribe and uses a global repository to store metadata about the available devices. The repository works with a monitor that is responsible to update the devices states and their QoS attributes. The sensor search is made by keywords and is sorted according to the QoS attributes prioritized by the user. Kothari *et al.* [41] presents an architecture-denominated DQS-Cloud to optimize the sensor search, provide resilience to faults and QoS degradation and also optimize system performance managing sensor data streams. The sensor search is based on keywords and considers the QoS attributes specified by users. Moreover, in order to reduce communication overhead, the authors proposes an optimization mechanism to reuse sensors flows to similar requests. The results showed that the optimization module is able to reduce the bandwidth and processing rate of the providers.

Shah *et al.* [11] presents a search mechanism based on coordinate virtual system to find process in P2P networks. A coordinate is assigned to a node representing a physical location in relation to other nodes. The sensor search uses keywords, and the returned sensors are ranked according to the euclidean distance to the QoS attributes specified by the user. A qualitative approach shows that the proposed search mechanism was the only one able to perform a precision query at real time. Ruta *et al.* [42] proposes a framework to manage semantic notations of data streams, devices, high level events and services. The requests uses the CoAP protocol based on the RESTful architectural style, which allow to use inference to support the sensor search and their compositions. A data mining mechanism was used to retrieve the sensor search in real time to improve the sensor selection. The sensor selection is based in the Concept Covering inference followed by a ranking algorithm.

Perera *et al.* [12] proposes a framework named CASSARAM, which performs the sensor search and selection regarding the QoS attributes specified by the user. The selection process is divided in two phases. In the first phase, the static sensor attributes, such as manufacturer or type, are used to limit the user space search. In the second phase, the result query of the first phase is evaluated in a multi-dimensional space where each axis corresponds to a QoS user attribute. The sensors are indexed regarding the comparative priority-based weighted index and ranked according to their euclidean distance to the optimal point. The authors also proposes a heuristic named comparative priority-based heuristic filtering, which removes the sensors that are far from the ideal point prioritizing the TOP-K selection. The results shows that using up to 10 000 sensors, the framework presents a satisfactory performance with a high precision.

Gao *et al.* [13] proposes the automated complex event implementation system to manages different run-time data streams. The sensors and their data streams are described according to SSN ontology and stored in a repository with their QoS attributes. The system acts such as a middleware between a sensor data stream and an application. The middleware are able to perform the sensor search and select using the Simple-Additive-Weighting algorithm to find the best trade-off between the specified QoS attributes.

Table III summarizes the main characteristics of the works presented in this Section. The authors [35]–[38] uses prediction models and do not consider QoS attributes to choose the sensors as they are just interesting in the sensor state. Ding *et al.* [39] also do not consider QoS attributes but worry

Table III. Related works summary.

Paper	Search technique	Search	Selection method	Quality of service
[35]	Prediction model	Sensor state	Single-period and multi-period Prediction model	No
[36]	Prediction model	Sensor state	Aggregated prediction model	No
[37]	Prediction model	Sensor state	Time-independent prediction model	No
[38]	Prediction model	Context information	Weighted slope one algorithm	No
[39]	Index	Keywords, sensor state Context information	B+ tree and R tree	No
[40]	Score and ranking	Keywords	Not specified	Yes
[41]	score, ranking and similarity	Keywords	Not specified	Yes
[42]	Inference and ranking	Keywords	Concept covering	No
[11]	Score and ranking	Keywords	Euclidean distance	Yes
[12]	Score and ranking	Context information	Euclidean distance and comparative priority-based heuristic filtering	Yes
[13]	Score and ranking	Context information	Simple additive weighting	Yes

about to offer efficient data structures to store the sensor state at run-time. On the other hand, [40]–[41] highlight the importance to select the sensors based on their QoS properties, but do not present a specific method for sensor search and selection. Authors [11]–[13] use methods to score and rank their sensors. The authors of [11] and [12] use the Euclidean distance of the sensor to the optimal point to score and rank while [13] applies the SAW method.

Briefly, these works present different mechanisms to perform the sensor search and selection considering QoS properties. However, they do not evaluate and compare the quality of the proposed solutions of a specific technique according to the number of desired QoS properties. Thus, to fulfill this gap, we have proposed a methodology to enable the comparison of different sensor search techniques from the quality of search perspective. In addition, a case study considering three sensor search techniques is presented to demonstrate our methodology.

6. CONCLUSION

The integration of IoTs with cloud computing composes the CoT and poses several new challenges. One of these challenges focus on the sensor search and selection field regarding a set of context properties explicit desired by an user. In this paper, we examined some multi-objective decision analysis methods applied in different scenarios. Specifically, we have analysed the SAW method, the TOPSIS and VIKOR according to the number of desired solutions and the number of context properties.

We described a methodology that uses the *Pareto* optimality concept to enable comparisons between the solutions proposed by these methods. The gathered results allowed to observe the impacts of 'the number of context properties' and 'the number of desired sensors' towards the quality of the final solution based on the *Pareto*-optimality concept.

Using the ONVGR, we observed when six, five and four context-properties are used, the proportion of optimal sensors retrieved is low because of the high number of context-properties conflicts, and also is practically the same for all. In contrast, when three or two context-properties are used, the proportion of optimal sensors retrieved increases significantly because of the small number of context properties conflicts.

In addition, the ONVGR value is directly proportional to the number of selected sensors. In other words, as more sensors are selected, more optimal sensors are used. Regarding the analysed MCDA methods, we could observe that the SAW presented the solution with better quality as their ONVGR value and number of fronts is equal or better than the values retrieved by TOPSIS and VIKOR. For future work, we will apply these methods in existent centralized and decentralized architectures for IoT where their performance will be measured. Also, new performance metrics and approaches for sensor selection will be analysed for each kind of architecture.

REFERENCES

1. Evans D. *The internet of things – how the next evolution of the internet is changing everything. Disponível em*, April 2011. (Available from: http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf) [Accessed on 23/10/2013].

2. Aazam M, Huh EN. Fog computing and smart gateway based communication for cloud of things. *2014 International Conference on Future Internet of Things and Cloud (FiCloud)*, 2014; 464–470.

3. Dada A, Thiesse F. Sensor applications in the supply chain: the example of quality-based issuing of perishables. *The Internet of Things*, Springer Science + Business Media, 2008; 140–154.

4. Atzori L, Iera A, Morabito G. The internet of things: a survey. *Computer Networks* 2010; **54**(15):2787–2805.

5. Islam SMR, Kwak D, Kabir MH, Hossain M, Kwak K. The internet of things for health care: a comprehensive survey. *IEEE Access* 2015; **3**:678–708.

6. Robles T, Alcarria R, Martin D, Morales A, Navarro M, Calero R, Iglesias S, Lopez M. An internet of things-based model for smart water management. *2014 28th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, 2014; 821–826.

7. Manna S, Bhunia SS, Mukherjee N. Vehicular pollution monitoring using IoT. *Recent Advances and Innovations in Engineering (ICRAIE), 2014*, 2014; 1–5.

8. Gerla M, Lee EK, Pau G, Lee U. Internet of vehicles: from intelligent grid to autonomous cars and vehicular clouds. *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014; 241–246.

9. Gubbi J, Buyya R, Marusic S, Palaniswami M. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems* 2013; **29**(7):1645–1660. DOI: http://dx.doi.org /10.1016/j.future.2013.01.010.

10. Bhuyan B, Sarma HKD, Sarma N, Kar A, Mall R. Quality of service (QoS) provisions in wireless sensor networks and related challenges. *Wireless Sensor Network* 2010; **2**(11):861–868. DOI: 10.4236/wsn.2010.211104. Available from: http://dx.doi.org/10.4236/wsn.2010.211104.

11. Shah M, Sardana A. Searching in internet of things using VCS. *Proceedings of the First International Conference on Security of Internet of Things*, SecurIT '12, ACM, New York, NY, USA, 2012; 63–67.

12. Perera C, Zaslavsky A, Christen P, Compton M, Georgakopoulos D. Context-aware sensor search, selection and ranking model for internet of things middleware. *Proceedings of the 2013 IEEE 14th International Conference on Mobile Data Management*, MDM '13, vol. 1, IEEE Computer Society, Washington, DC, USA, 2013; 314–322.

13. Gao F, Ali M I, Mileo A. Semantic discovery and integration of urban data streams. *Ceur Workshop Proceedings*, Vol. 1280, 2014; 15–30.

14. Vermesan O, Friess P, Guillemin P, Gusmeroli S, Sundmaeker H, Bassi A, Jubert IS, Mazura M, Harrison M, Eisenhauer M, *et al.* Internet of things strategic research roadmap. *Internet of Things-Global Technological and Societal Trends* 2011; **1**:9–52.

15. Zhang M, Sun F, Cheng X. Architecture of internet of things and its key technology integration based-on RFID. *2012 Fifth International Symposium on Computational Intelligence and Design*, Institute of Electrical & Electronics Engineers (IEEE), 2012. DOI: 10.1109/iscid.2012.81. Available from: http://dx.doi.org/10.1109/ISCID.2012.81.

16. Khan R, Khan SU, Zaheer R, Khan S. Future internet: the internet of things architecture, possible applications and key challenges. *2012 10th International Conference on Frontiers of Information Technology*, 2012; 257–260. DOI: 10.1109/FIT.2012.53.

17. Fersi G. Middleware for internet of things: A study. *2015 International Conference on Distributed Computing in Sensor Systems*, 2015; 230–235. DOI: 10.1109/DCOSS.2015.43.

18. Perera C, Vasilakos AV. A knowledge-based resource discovery for internet of things. *Knowledge Based Systems* 2016; **109**:122–136. DOI: http://dx.doi.org/10.1016/j.knosys.2016.06.030. Available from: http://www.sciencedirect. com/science/article/pii/S0950705116302015.

19. Aberer K, Hauswirth M, Salehi A. Middleware support for the internet of things. *Proceedings of 5. GI/ITG KuVS FachgespraechDrahtlose Sensornetze*, 2006; 15–19.

20. Soldatos J, Serrano M, Hauswirth M. Convergence of utility computing with the internet-of-things. *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, 2012; 874–879.

21. Xu L, Yang JB. Introduction to multi-criteria decision making and the evidential reasoning approach, Working Paper Series No. 0106, Manchester School of Management, University of Manchester Institute and Technology, 2001.

22. Dodgson JS, Spackman M, Pearman A, Phillips LD. *Multi-Criteria Analysis: A Manual*. Department for Communities and Local Government: London, 2009.

23. Tzeng GH, Huang JJ. *Multiple Attribute Decision Making: Methods and Applications*, A Chapman & Hall book. Taylor & Francis, 2011.

24. Memariani A, Amini A, Alinezhad A. Sensitivity analysis of simple additive weighting method (SAW): the results of change in the weight of one attribute on the final ranking of alternatives. *Journal of Industrial Engineering* 2009; **4**:13–18.

25. Abdullah L, Adawiyah CR. Simple additive weighting methods of multi criteria decision making and applications: A decade review. *International Journal of Information Processing and Management* 2014; **5**(1):39–49.

26. Afshari A, Mojahed M, Yusuff RM. Simple additive weighting approach to personnel selection problem. *International Journal of Innovation, Management and Technology* 2010; **1**(5):511–515.

27. Behzadian M, Otaghsara SK, Yazdani M, Ignatius J. A state-of the-art survey of TOPSIS applications. *Expert Systems with Applications* 2012; **39**(17):13051–13069.
28. Opricovic S, Tzeng GH. Compromise solution by mcdm methods: a comparative analysis of vikor and topsis. *European Journal of Operational Research* 2004; **156**(2):445–455.
29. Huang JJ, Tzeng GH, Liu HH. A revised VIKOR model for multiple criteria decision making - the perspective of regret theory. *Communications in Computer and Information Science*, Springer Science + Business Media, 2009; 761–768.
30. Marler RT, Arora JS. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization* 2004; **26**(6):369–395.
31. Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation* 2002; **6**(2):182–197.
32. Collette Y, Siarry P. *Multiobjective Optimization*. Springer Berlin Heidelberg, 2004. DOI: 10.1007/978-3-662-08883-8. Available from: http://dx.doi.org/10.1007/978-3-662-08883-8.
33. Perera C, Zaslavsky A, Christen P, Georgakopoulos D. Context aware computing for the internet of things: a survey. *IEEE Communications Surveys Tutorials* 2014; **16**(1):414–454.
34. Romer K, Ostermaier B, Mattern F, Fahrmair M, Kellerer W. Real-time search for real-world entities: a survey. *Proceedings of the IEEE* 2010; **98**(11):1887–1902.
35. Elahi BM, Romer K, Ostermaier B, Fahrmair M, Kellerer W. Sensor ranking: a primitive for efficient content-based sensor search. *International Conference on Information Processing in Sensor Networks, 2009. IPSN 2009*, 2009; 217–228.
36. Ostermaier B, Romer K, Mattern F, Fahrmair M, Kellerer W. A real-time search engine for the web of things. *Internet of Things (IOT), 2010*, 2010; 1–8.
37. Truong C, Romer K. Content-based sensor search for the web of things. *2013 IEEE Global Communications Conference (GLOBECOM)*, 2013; 2654–2660.
38. Carlson D, Schrader A. Ambient ocean: a web search engine for context-aware smart resource discovery. *Internet of Things (iThings), 2014 IEEE International Conference on, and Green Computing and Communications (GreenCom), IEEE and Cyber, Physical and Social Computing(CPSCom)*, IEEE, 2014; 177–184.
39. Ding Z, Dai J, Gao X, Yang Q. A hybrid search engine framework for the internet of things. *2012 ninth Web Information Systems and Applications Conference (WISA)*, 2012; 57–60.
40. Guinard D, Trifa V, Karnouskos S, Spiess P, Savio D. Interacting with the soa-based internet of things: discovery, query, selection, and on-demand provisioning of web services. *IEEE Transactions on Services Computing* 2010; **3**(3):223–235.
41. Kothari A, Boddula V, Ramaswamy L, Abolhassani N. Dqs-cloud: a data quality-aware autonomic cloud for sensor services. *2014 International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, 2014; 295–303.
42. Ruta M, Scioscia F, Pinto A, Sciascio ED, Gramegna F, Ieva S, Loseto G. Resource annotation, dissemination and discovery in the semantic web of things: A coap-based framework. *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, 2013; 527–534. DOI: 10.1109/GreenCom-iThings-CPSCom.2013.103.