# Lightweight network interface selection for reliable communications in multi-technologies wireless sensor networks

Brandon Foubert, Nathalie Mitton

▶ **To cite this version:**

Brandon Foubert, Nathalie Mitton. Lightweight network interface selection for reliable communications in multi-technologies wireless sensor networks. DRCN 2021 - 17th International Conference on the Design of Reliable Communication Networks, Apr 2021, Milan, Italy. hal-03141833

**HAL Id: hal-03141833**

**https://hal.inria.fr/hal-03141833**

Submitted on 15 Feb 2021

# Lightweight network interface selection for reliable communications in multi-technologies wireless sensor networks

Brandon Foubert and Nathalie Mitton
*Inria* - Lille, France
{brandon.foubert, nathalie.mitton}@inria.fr

*Abstract*—**Wireless sensor networks (WSN) are composed of hardware constrained and battery-powered devices that communicate wirelessly. WSN find more and more applications, but their deployment is limited among others by the range and the throughput of the communication technology used. Several technologies are available nowadays, with various performances, cost and coverage. One solution to overcome the deployment limitations and in some cases extend the coverage would be to dynamically select the technology based on the data requirements, environment, geographic location, etc. Thus we need multi-technologies WSN devices and efficient algorithms to select the best available technology in an autonomous and local way. This issue is known as Network Interface Selection (NIS). Multi-Attribute Decision Making (MADM) methods are an efficient tool to tackle NIS. Among MADM methods is Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS). However, TOPSIS suffers from a rank reversal issue, which may alter the ranking quality. Furthermore, TOPSIS method is computationally heavy, which might increase the energy consumption of the constrained devices and the latency of the network. In this paper, we introduce a lightweight TOPSIS-based method tailored for NIS in WSN, allowing more reliable communications. Experimental results obtained on real hardware, *i.e.*, Pycom FiPy modules, show an improvement in computation time of $38\%$ while maintaining a selection similar to TOPSIS in $82\%$ of runs.**

*Index Terms*—**WSN, NIS, TOPSIS, MADM, rank reversal**

## I. INTRODUCTION

Wireless Sensor Networks (WSN) are a useful tool for many applications, such as environmental monitoring [1]. For the sake of flexibility and simplicity, nodes that compose a WSN generally run on batteries, communicate wirelessly and are not directly manned by a remote operator. Thus, those nodes have to behave autonomously in a way that saves as much energy as possible. Nodes collect data using sensors, and share that data with neighbor nodes using a wireless communication technology. Therefore, WSN deployments are constrained by the limits of the technology used by the nodes, in terms of coverage and throughput. Moreover, some technologies are so constrained that they may not be able to comply with specific data requirements such as data flow differentiation. Besides, outdoor nodes have to bear the weather changes (*e.g.,* rain) which impacts the quality of the wireless links.

Nowadays, several technologies are available, and each one offers different performances and characteristics. An interesting solution to the aforementioned limitations would be to deploy nodes equipped with several wireless technologies [2].
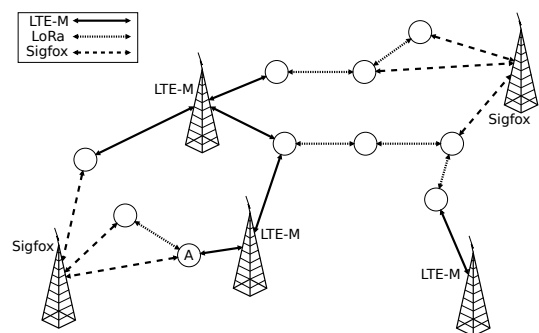


Fig. 1. Multi-technologies WSN.

Thereby, nodes would have the possibility to use the best technology at their disposal. The choice can be influenced by several factors and must be driven by the environment, application and data requirements and constraints. As an example, let us consider the multi-technologies WSN depicted in Figure 1. In this network, nodes periodically collect and send environmental monitoring data wirelessly. For such a type of communication, node A will use a low power and low throughput technology to save up energy, such as Sigfox. However, upon detection of an unexpected and important event (*e.g.,* risk of hail), node A may need to trigger an alarm to be sent to an operator. In this case, a faster technology, such as LTE-M, is needed to warn the operator as quickly as possible while ignoring other criteria such as energy consumption. For such cases, there is a need to smartly switch the transmission technology based on data priorities to increase the communication's reliability. To that end, nodes have to use specific methods to autonomously choose which technology is the best suited depending on the data requirements. This issue is known as Network Interface Selection (NIS).

Several tools are available in the literature to tackle the NIS problem, among them are the Multiple Attribute Decision-Making (MADM) methods. MADM methods provide a ranking of different alternatives based on their attributes and their

TABLE I
MADM DECISION MATRIX

|       | $P_1$ | $P_2$ | ... | $P_m$ |
|-------|-------|-------|-----|-------|
|       | $w_1$ | $w_2$ | ... | $w_m$ |
| $A_1$ | $x_{11}$ | $x_{12}$ | ... | $x_{1m}$ |
| $A_2$ | $x_{21}$ | $x_{22}$ | ... | $x_{2m}$ |
| ...   | ... | ... | ... | ... |
| $A_n$ | $x_{n1}$ | $x_{n2}$ | ... | $x_{nm}$ |

associated weights. One of the most used and studied MADM methods is Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS). Said simply, TOPSIS offers to compare alternatives based on their mathematical distances to two ideal positive and negative alternatives.

However, TOPSIS suffers from an issue known as rank reversal. A rank reversal happens when the ranking is modified following the removing of one of the alternatives under study. This can alter the quality of the ranking and lead to a sub-optimal NIS. In our case, this could outcome in too many useless and costly technology switches. Moreover, considering hardware constrained WSN nodes, TOPSIS computation is resource-intensive. This would decrease the devices' lifetime and may overload the devices' limited memory which leads to hardware failure. We address those issues in this paper, by proposing a lightweight TOPSIS-based NIS method optimized for WSN devices. Our method simplifies TOPSIS computations and completely eliminates rank reversal by modifying the TOPSIS normalization algorithm. This results in less complexity and provides time and energy savings. The pertinence and performances of our approach are assessed via experimentation on real hardware multi-technologies WSN devices, FiPy modules from Pycom. Results show an improvement in the computation time of around $38\%$, which in turn results in energy savings, while the technology selection is equivalent to using the classic TOPSIS method in $82\%$ of the experiments.

The rest of this paper is organized as follows: Section II presents the background about MADM and TOPSIS, and Section III explains what issues have to be faced with TOPSIS. Then Section IV introduces the existing related work about those issues. In Section V, we detail our contribution and in Section VI we present our experiments and the results we have obtained. Finally, Section VII concludes this work.

## II. MADM AND TOPSIS BACKGROUND

Multi-technologies devices have to autonomously select the best communication technology based on many factors. In the literature, several tools are available to perform this Network Interface Selection (NIS): utility and cost functions, Markov chains, fuzzy logic, game theory, data mining, Dempster-Shafer theory, to name a few. Particularly, Multi-Attribute Decision Making (MADM) methods [3] are commonly used for NIS. MADM methods are interesting as they rank several alternatives, based on their attributes as well as the relative importance associated to those attributes.

The problem can be modelled with a decision matrix as shown in Table I. It is composed of $A = \{A_i \mid i = 1, 2, ..., n\}$

the set of the alternatives, $P = \{P_j \mid j = 1, 2, ..., m\}$ the set of the attributes and $W = \{w_j \mid j = 1, 2, ..., m\}$ the set of the weights associated to each attribute. Applied to NIS, $A$ is the set of technologies, $P$ the set of attributes associated to those and $W$ the data requirements. The MADM methods take as input a decision matrix and output a ranking of the alternatives. Several MADM methods exist, the most known being: Simple Additive Weighting (SAW), Weighting Product (WP), Analytical Hierarchy Process (AHP) and Gray Relational Analysis (GRA).

One of the most used and studied methods is Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) [4]. TOPSIS ranks alternatives depending on their relative mathematical distance to the ideal solution. The TOPSIS method runs the following steps:

1) The values $x_{ij}$ of each attribute from the decision matrix (cf. Table I) are normalized according to Equation 1.

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{n} x_{ij}^2}} \tag{1}$$

2) The normalized values $r_{ij}$ are weighted according to Equation 2.

$$v_{ij} = w_j r_{ij}, \ \sum_{j=1}^{m} w_j = 1 \tag{2}$$

3) The positive and negative ideal alternatives $A^+$ and $A^-$ are constructed according to Equation 3.

$$A^+ = [v_1^+ ... v_m^+] \\ A^- = [v_1^- ... v_m^-] \tag{3}$$

4) The attribute values of the ideal alternatives are determined according to Equation 4 for upward attributes (*e.g.* range) or Equation 5 for downward attributes (*e.g.* latency).

$$v_j^+ = Argmax\{v_{ij}, i = 1, ..., n\} \\ v_j^- = Argmin\{v_{ij}, i = 1, ..., n\} \tag{4}$$

$$v_j^+ = Argmin\{v_{ij}, i = 1, ..., n\} \\ v_j^- = Argmax\{v_{ij}, i = 1, ..., n\} \tag{5}$$

5) The distances between each alternative and the positive and negative ideal alternatives $A^+$ and $A^-$ are computed according to Equation 6.

$$S_i^+ = \sqrt{\sum_{j=1}^{m}(v_j^+ - v_{ij})^2} \\ S_i^- = \sqrt{\sum_{j=1}^{m}(v_j^- - v_{ij})^2} \tag{6}$$

6) Finally, the relative closeness to the ideal solution is computed for each alternative according to Equation 7 and a ranking is established based on those values.
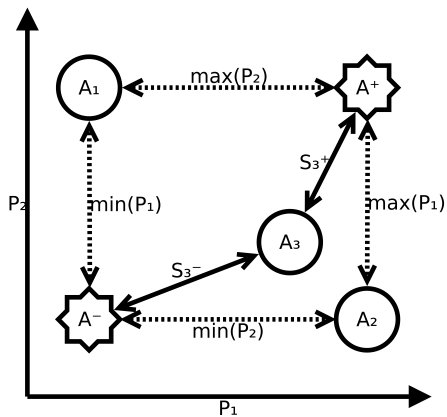
$$C_{TOPSIS} = \frac{S_i^-}{S_i^- + S_i^+} \tag{7}$$

Fig. 2. Representation of TOPSIS with three alternatives and two attributes.

| | $P_1$ | $P_2$ | $P_3$ |
|---|---|---|---|
| $A_1$ | 1.024537 | 7.828443 | 8.650221 |
| $A_2$ | 4.226149 | 0.09865402 | 4.673396 |
| $A_3$ | 8.026353 | 5.455392 | 2.536936 |
| $A_4$ | 1.700537 | 1.398855 | 0.7656412 |

When using TOPSIS for NIS, the technology with the highest value of $C_{TOPSIS}$ is selected. A graphical representation of the TOPSIS method with three alternatives and two attributes is depicted in Figure 2.

## III. TOPSIS PROBLEM STATEMENT

TOPSIS is particularly interesting, as it grades alternatives based not only on the closeness from the best alternative but also on the distance from the worst one. However, TOPSIS suffers from an issue known as rank reversal that can happen when a non-optimal alternative is removed from the ranking. This can alter the quality and pertinence of the ranking. Rank reversal is an issue common to several MADM methods. With an ideal method, the ranking of alternatives should not be altered when another alternative is removed. The cause of rank reversal is the normalization algorithm. Indeed, the TOPSIS normalization (a.k.a. euclidean normalization) computes the normalized values for an attribute based on the values of all the other alternatives for that same attribute. Thus if set $A$ changes, the result of Equation 1 also changes, which may modify the final ranking.

To clarify rank reversal let us consider an example. Table II represents a simple decision matrix randomly filled. Running TOPSIS on it outputs a ranking order corresponding to $[A_1, A_3, A_2, A_4]$. If the alternative $A_4$ was to be removed from the ranking (*e.g.* because of a broken link for example), it is expected that the ranking of the remaining alternatives should not be altered and therefore should correspond to $[A_1, A_3, A_2]$. However, running TOPSIS on Table II after removal of alternative $A_4$ outputs a ranking corresponding to $[A_3, A_1, A_2]$. This corresponds to a rank reversal. Applied to NIS, it means that the loss of the wireless link of technology $A_4$ would change the selected technology from $A_1$ to $A_3$. This would cause a technology switch which will require energy and does not bring any overall improvement.

It is to be noted that rank reversal is not a theoretical issue for multi-technologies WSN devices. Actually, the wireless technologies' links' quality depends on many factors such as atmospheric and environmental conditions, which vary heavily

across the year. This may result in broken links, thus removing a technology from the set of alternatives and potentially resulting in rank reversal, as seen in the previous example. The frequency of such events is entirely dependent on external factors and cannot be anticipated, thus links' quality has to be considered in the NIS process. Rank reversal could lead to the selection of a sub-optimal technology, on top of spending energy for switching between technologies.

A second issue posed by TOPSIS-based NIS on constrained devices is the complex computations that are required. TOPSIS method as seen in Section II is based on computations that use numerous operations and memory accesses. WSN devices are generally hardware constrained, energy-limited and a repetitive execution of the TOPSIS method will have a considerable impact on the energy consumption of nodes. As an example, the Pycom FiPy's CPU [5] holds two cores that can go up to 240 MHz. A classic laptop CPU, *e.g.*, the Intel® Core™ i7-8650U, holds four cores that can go up to 4.20 GHz.

## IV. RELATED WORK

Several works have been conducted to mitigate rank reversal in TOPSIS or to apply TOPSIS to NIS. [6] proposes an iterative TOPSIS method, where TOPSIS is executed, then the worst alternative is removed from the ranking, and TOPSIS is re-executed, as long as there is more than one alternative in the ranking. The remaining one is selected as communication technology. [7] combines TOPSIS with fuzzy logic, in order to improve how uncertain attributes are taken into account. [8] introduces alternative methods based on TOPSIS, but with different normalization algorithms using maximum and minimum values of the attributes. [9] compares several NIS methods applied to heterogeneous WSN. [10] introduces an original MADM method along with an in-depth analysis of TOPSIS. [11] proposes a new Service-based Interface Selection Scheme algorithm based on TOPSIS to enable NIS applied to vehicle-to-vehicle communications scenarios. [12] details a fast TOPSIS-based NIS technique for vertical handover in heterogeneous emergency communication systems.

Overall, those propositions reduce the probability of occurrence of rank reversal, but does not nullify it because the euclidean normalization is still used. Furthermore, some of the proposed modifications tend to increase the complexity of the TOPSIS method. This would increase the execution time of TOPSIS and in turn the energy consumption of the nodes, thus reducing their lifetime.

To the extent of the authors knowledge, no works have been conducted to propose a rank reversal free TOPSIS-based method for NIS specifically for energy constrained devices.

Thus, in this in paper we introduce a lightweight TOPSIS-based NIS method that aims not only to reduce the complexity and energy consumption of TOPSIS, but also to completely eliminate rank reversal.

## V. LIGHTWEIGHT TOPSIS

As stated in Section III, the rank reversal issue is due to TOPSIS' normalization which computes normalized values based on all the other alternatives' values. Moreover, this normalization method is rather complex, and may increase the energy consumption of nodes.

Thus, we propose to use a simplified normalization method, which will not cause rank reversal and simplify the computations. Rank reversal happens because other alternatives are taken into account when computing normalized values. Thus, our proposition is to compute those values without taking into account other alternatives values. Therefore, we need a stable normalization referential to measure our values against. We know that multi-technologies devices have a fixed set of technologies available. Those are not supposed to change after deployment, and they have fixed maximum and minimum capabilities. We propose to use those maximum and minimum bounds as referential for our normalization.

### A. Algorithm

---
**Algorithm 1** Lightweight normalization
---
**Require:** $x_{ij}$ the raw value of each attribute $j$ for each candidate $i$
  **for** each attribute $P_j$ **do**
    **if** $P_j$ is an upward attribute **then**
      $B_j^+$ is the upper bound of $P_j$
      $r_{ij} = \frac{x_{ij}}{B_j^+}$
    **else if** $P_j$ is a downward attribute **then**
      $B_j^-$ is the lower bound of $P_j$
      $r_{ij} = \frac{B_j^-}{x_{ij}}$
    **end if**
  **end for**
  **return** $r_{ij}$ the normalized value of $x_{ij}$

---

That simplification takes the form of Algorithm 1, which replace Equation 1 in the steps of our lightweight TOPSIS. Each value $x_{ij}$ is normalized by being divided with the upper or the lower bound of its attribute $j$. Upward attributes values are divided by their upper bound, while downward attributes divide their lower bound. The set $B = \{B_j^+, B_j^- \mid j = 1, 2, ..., m\}$ is composed of the upper and lower bounds of each attribute $j$, such that $\forall \, x \in B, \, 0 < x < +\infty$. $B$ is stable, thus normalized values from the alternatives will not be altered by the removing of any other alternative. This completely eliminates rank reversal and reduces algorithmic complexity at once. Indeed, Equation 1 requires the computation of the denominator $\sqrt{\sum_{i=1}^{n} x_{ij}^2}$ for each value of $j$ (for $m$ attributes). This is not required with our bounded normalization and only the division between the bound and the value is computed.

Knowing the fixed bounds allows us to simplify TOPSIS further: Equation 3 is used to establish the ideal positive and negative alternatives. Extreme values are found according to Equation 4 for upward attributes or Equation 5 for downward ones. Those operations require many comparisons. With bounded normalization, we can simplify the determination of the ideal alternatives: determination of $A^+$ and $A^-$ is trivial, as the normalized maximum and minimum bounds of the attributes are respectively equals to 1 and 0. Thus, Equations 4 and 5 can be simplified by Equation 9. In turn, determination of the ideal alternatives according Equation 3 shows that these are static and shown in Equation 8. Finally, distances computation according to Equation 6 can be simplified by Equation 10. Indeed, as the ideal alternatives are known and static, we thus know that $v_j^+ = 1$ and $v_j^- = 0$.

Those simplifications reduce the complexity of the TOPSIS method. Moreover, as the normalization uses a stable referential, rank reversal probability is eliminated. Those modifications thus reduce the time required for execution, as we will see in Section VI.

$$
\begin{aligned}
A^+ &= [1...1] \\
A^- &= [0...0]
\end{aligned} \tag{8}
$$

$$
\begin{aligned}
v_j^+ &= 1 \\
v_j^- &= 0
\end{aligned} \tag{9}
$$

$$
\begin{aligned}
S_i^+ &= \sqrt{\sum_{j=1}^{m}(1 - v_{ij})^2} \\
S_i^- &= \sqrt{\sum_{j=1}^{m} v_{ij}^2}
\end{aligned} \tag{10}
$$

### B. Complexity

The reduced complexity of our algorithm can be assessed with an algorithmic complexity comparison. As big $O$ notation is only pertinent for large inputs, we choose to quantify the number of operations spared with our method instead of classic TOPSIS. We consider one operation as one of the four basic arithmetic operations: addition, subtraction, multiplication and division. We also consider square root and value comparison as a single operation. This is just an estimation and is not exact as a square root is decomposed into multiple simpler operations when computed. However, as the exact decomposition is dependent on the hardware, it is irrelevant to assign a precise operation cost to a square root. Hereafter we consider $n$ and $m$ to be the dimensions of the decision matrix.

Firstly, Equation 1 requires at least $3nm$ operations, while using Algorithm 1 instead reduces it to $nm$ operations. Replacing Equations 3, 4 and 5 by Equations 8 and 9 spares the cost of the min-max algorithm, thus $2(mn - 1)$ operations. Finally, using Equation 10 instead of Equation 6 spares $nm$ operations. Our proposition thus spares a total of $5mn - 2$ operations.
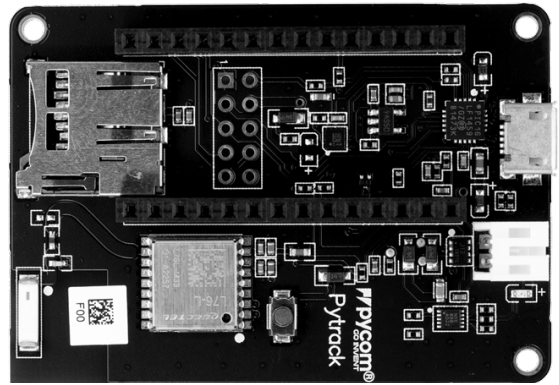
Fig. 3. FiPy board from Pycom [13].



Fig. 4. Pytrack sensor shield from Pycom [13].

TABLE III
ATTRIBUTES' WEIGHTS

|  | Energy | Delay | Cost |
|---|---|---|---|
| $W_{monitoring}$ | 0.6 | 0.1 | 0.3 |
| $W_{alarm}$ | 0.1 | 0.8 | 0.1 |

## VI. EXPERIMENTS

We implemented both algorithms in MicroPython on FiPy modules from Pycom, coupled with Pytrack expansion boards. Both are depicted in Figures 3 and 4. Those devices offer five different wireless communication technologies, and provide hardware close to the one used in WSN. The available technologies on FiPy platform are WiFi, LoRa, Sigfox, LTE-M, NB-IoT and Bluetooth Low Energy. Each one comes with different performances, based on different metrics such as: energy consumption, economical cost, throughput, delay, loss rate, etc. Attributes of each technology are used to fill the decision matrix values $x_{ij}$ used as input for the NIS algorithms. Weights associated to attributes are determined based on the data requirements. Table III shows an example set of weights that could be used: for regular monitoring data the weight and thus importance of the energy consumption will be higher. This would probably lead to an NIS of the best energy-efficient technology (*e.g.,* Sigfox). On the contrary, for an alarm the weight of delay will be higher, leading to an NIS of the fastest technology (*e.g.,* WiFi).

### A. Rank reversal prevalence

We wanted to know how painful can be a rank reversal using TOPSIS for NIS. We ran experiments to quantify the prevalence of rank reversal using TOPSIS. The nodes execute the following steps: *i)* create a random matrix, *ii)* run TOPSIS on it and *iii)* compute the resulting ranking. Then we randomly remove one of the potential alternative and the new ranking was computed. TOPSIS was run again on the matrix without the alternative removed from the ranking, and the resulting ranking was compared with the previous ranking. If the order of remaining options was different, then a rank reversal happened.

Results are highly dependent on the size of the matrices. Generally, the bigger the decision matrix, the more rank reversals as we can see in Figure 5. Large matrices are not a current realistic representation of NIS in WSN. Multi-technologies WSN nodes have several technologies available, but it is very unlikely that plain nodes carry hundreds of technologies. Similarly, technologies can have tens of attributes compared, but it is unlikely to be hundreds. Nonetheless, later, hardware will integrate more and more computing resources and communication technologies so our proposition will be able to scale with them. Still, we can see that even with small $(5 \times 5)$ matrices as we can obtain with FiPy modules, rank reversal happens approximately in $30\%$ of the experiments. Rank reversal may cause useless technology switches, that are costly energy-wise. Larger matrices imply more frequent rank reversal, which emphasizes the need for a solution as ours. This is considerable if we assume TOPSIS to be run periodically to select the best technology after attributes or data requirements change.

### B. Proposition evaluation

We compare the performances of a classic TOPSIS with our lightweight TOPSIS. We measure the time needed for the algorithms completion with the Timer library available for the FiPy as well as the similarity between the resulting NIS. It is worth noting that TOPSIS does not embed an objective comparison referential to estimate the quality of a ranking. However, TOPSIS is considered to produce a good quality ranking and is thus commonly used as a point of comparison. The obtained results are visible in Figure 6.

We obtain a mean speed up of the computing time of $38\%$. At the same time, we still maintain a similarity with TOPSIS ranking in $82\%$ of the experiments. Note that the ranking in the remaining $18\%$ of the experiments cannot be qualified as worse for all cases since it mainly depends on the application and of what is expected or required. The ranking is only different from TOPSIS' ranking, which we used as a reference, but is not a ground truth. If we look at what we obtained when using a $(5 \times 5)$ matrix for a population of 7000 experiments with the results rounded to two decimal places, the mean
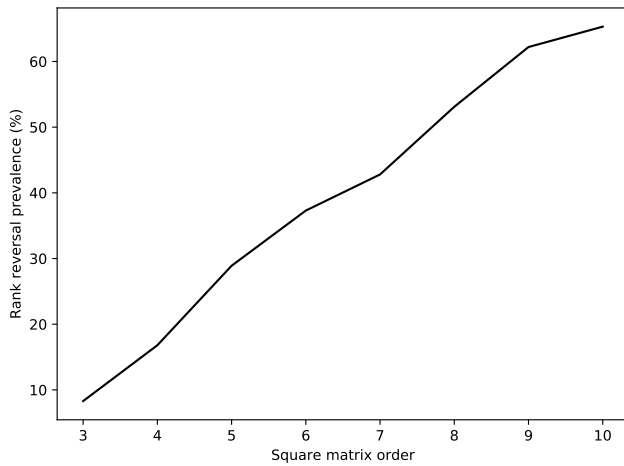
Fig. 5. Rank reversal prevalence as a function of the decision matrix' size.



Fig. 6. Classic and lightweight TOPSIS run times.

execution time of the classic TOPSIS is 4.79 ms, while the mean execution time of our lightweight TOPSIS is 2.96 ms. This means that a node could benefit from a mean time of 1.83 ms longer sleep periods between two TOPSIS executions. Based on the FiPy CPU data-sheet [5], with a maximum CPU consumption of 68 mA and a power supply of 3.6 V, it would save up to approximately 448 µJ per TOPSIS run. Data-sheets are notoriously optimistic, so in practice the energy savings could be even more significant. The standard deviation is of 0.05 ms, and the confidence intervals are $+/-2.76*10^{-3}$ ms and $+/-2.48*10^{-3}$ respectively for classic TOPSIS and for our lightweight TOPSIS, with a confidence level of 99.999%. Larger matrices offer similar results.

## VII. Conclusion

WSN are composed of energy constrained devices running on batteries. WSN deployment is constrained by the technology used by the nodes to communicate. To overcome this limitation, multi-technologies nodes could be designed. Those nodes should select the optimal communication technology based on the environment and the data requirements the nodes have to fulfill. MADM is an interesting multiple-criteria decision tool to compare several alternative and produce a ranking. The TOPSIS method especially proposes to rank alternatives not only on their resemblance with the best alternative but also on their dissimilarity with the worst alternative. However, TOPSIS is rather complex and suffers from rank reversal, which would increase the energy consumption of the nodes.

In this article, we introduce a lightweight TOPSIS method that reduces the complexity of the computations and eliminates rank reversal issues. It aims to respect energy and capabilities constraints commonly found in WSN. Experimental performance assessment of our proposition shows a shortening of computation time around 38%. The resulting NIS is still similar with the one obtained using classic TOPSIS in 82% of the experiments. This directly allow nodes to reduce their energy consumption, as they can sleep for longer periods.
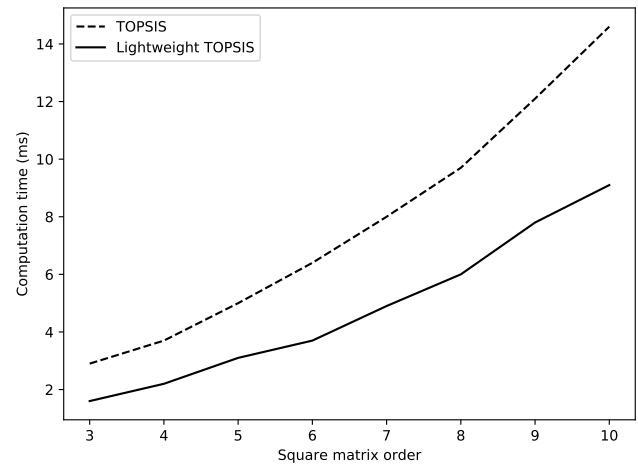
As future work, we are planning to extend our NIS method to select not only the technology used to communicate but also which neighbor. This will take into account the fact that each device has multiple technologies available for communication.

## References

[1] D. Ye, D. Gong, and W. Wang, "Application of wireless sensor networks in environmental monitoring," in *international Conference on Power Electronics and Intelligent Transportation System*, 2009.

[2] B. Foubert and N. Mitton, "Autonomous Collaborative Wireless Weather Stations: A Helping Hand for Farmers," *ERCIM News*, no. 119, pp. 37–38, 2019.

[3] R. V. Rao, *Decision Making in the Manufacturing Environment: Using Graph Theory and Fuzzy Multiple Attribute Decision Making Methods*. Springer, 2007, ch. Introduction to Multiple Attribute Decision-making (MADM) Methods, pp. 27–41.

[4] C. L. Hwang and K. Yoon, *Multiple Attribute Decision Making: Methods and Applications*. New York: Springer-Verlag, 1981.

[5] *ESP32 Series Datasheet*, Espressif Systems, 2020, version 3.3.

[6] F. Bari and V. C. M. Leung, "Multi-attribute network selection by iterative topsis for heterogeneous wireless access," in *IEEE Consumer Communications and Networking Conference*, 2007.

[7] W. Zhang, "Handover decision using fuzzy madm in heterogeneous networks," in *IEEE Wireless Communications and Networking Conference*, 2004.

[8] M. A. Senouci, M. S. Mushtaq, S. Hoceini, and A. Mellouk, "Topsis-based dynamic approach for mobile network interface selection," *Computer Networks*, vol. 107, pp. 304 – 314, 2016.

[9] E. Stevens-Navarro and V. W. S. Wong, "Comparison between vertical handoff decision algorithms for heterogeneous wireless networks," in *IEEE Vehicular Technology Conference*, 2006.

[10] F. Bari and V. C. M. Leung, "Automated network selection in a heterogeneous wireless network environment," *IEEE Network*, vol. 21, no. 1, pp. 34–40, 2007.

[11] L. H. Teixeira and A. Huszak, "Preemptive network selection for v2v communication," in *International Conference on Telecommunications and Signal Processing*, 2019.

[12] I. Bisio and A. Sciarrone, "Fast multiattribute network selection technique for vertical handover in heterogeneous emergency communication systems," *Wireless Communications and Mobile Computing*, vol. 2019, pp. 1–17, 2019.

[13] "Pycom website," https://pycom.io/, [Online; accessed 26-May-2020].