10.18605/2175-7275/cereus.v16n3p437-449

<< Recebido em: 08/07/2024 Aceito em: 28/09/2024. >>

ARTIGO ORIGINAL

REVISTA CEREUS
ISSN: 2175-7275

An ILS Heuristic to Determine Coverage of a Region Using Wireless Sensor Network

Uma Heurística ILS para Determinar a Cobertura de uma Região Utilizando Rede de Sensores Sem Fio

Marcelo da Silva Lisboa[1] e Marcelo Lisboa Rocha[2]

ABSTRACT

Wireless sensor networks are basically composed of sensor nodes, communication interfaces and gateway nodes. The integration of these devices provides a connection between the real and the virtual world with their diverse sensing capabilities. A problem regarding wireless sensor networks is linked to the positioning of sensor nodes to cover a certain region. This problem fits into the group of problems that are difficult to optimize, requiring the use of high-performance algorithms. The iterative local search (ILS) method was used to solve the problem in question with local search techniques and appropriate perturbations. The results obtained were considered satisfactory, as better solutions were found for all the problems studied, at the cost of a small increase in computational time.

Keywords: Wireless Sensors Network. Coverage of a Region. ILS Heuristic.

RESUMO

Redes de sensores sem fio são compostas basicamente por nodos sensores, interfaces de comunicação e nodos gateway. A integração desses dispositivos proporciona uma conexão do mundo real com o virtual com sua capacidade de sensoriamento diversificada. Um problema relativo redes de sensores sem fio está ligado ao posicionamento dos nodos sensores para cobertura de uma determinada região. Esse problema se encaixa no grupo de problemas difíceis de serem otimizados, sendo necessária a utilização de algoritmos de alto desempenho. O método de busca local iterativa (ILS) foi utilizado para resolver o problema em questão com técnicas de busca local e perturbações adequadas. Os resultados obtidos foram considerados satisfatórios, pois em todos os problemas estudados foram encontradas melhores soluções, ao custo um pequeno aumento do tempo computacional.

Palavras-chave: Rede de Sensores sem Fio. Cobertura de uma Região. Heurística ILS.

[1] Bach. in Data Science. Information Technology Department - Secretary of Administration of Tocantins.

E-mail: marcelodasilvalisboa1@gmail.com

ORCID: 0009-0000-1526-8825

[2] DSc in Electrical Engineering. Postgraduate Program in Governance and Digital Transformation - UFT.

E-mail: mlisboa@uft.edu,br

ORCID: 0000-0002-4034-0021

1. INTRODUCTION

The demand for electronic devices is constantly growing. According to consultancy Gartner (2023): “worldwide semiconductor revenue is projected to reach US$624 billion, registering annual growth of 16.8% in 2024”. This demand grows along with man's need to stay informed and monitor the environment in which he lives. A great ally of man in meeting this need are Wireless Sensor Networks (WSN), which have attracted the attention of a number of researchers in recent years.

A WSN is a special type of mobile ad hoc network composed of independent and compact devices with sensing, processing and communication power that we can call sensor nodes, simply nodes or nodes. Elson and Estrin (2004) observed that the benefits obtained by a distributed network, which is characteristic of the WSN, are due to the integration of several sensors, which gives an overview of the world that an isolated sensor would not provide.

A WSN is formed by a certain number of nodes deployed in an area collaborating with each other through a wireless network. Some WSN applications will require hundreds or even thousands of nodes, which would be unfeasible with traditional cable networks, due to the complexity and cost of installation and maintenance. Since monitoring in volcanoes, seas, rivers, among others, is unfeasible and even impossible to install traditional cable networks.

A node in this network is mainly composed of a sensor and communication unit. You are responsible for monitoring the area, which may be temperature, light, pressure, humidity, among others, and may have more than one type of sensor in a node. The communication unit is responsible for transmitting and receiving data. Despite being qualified as a type of ad hoc network, WSNs have some differences in relation to traditional networks. Among these, energy restrictions, high node density, failure proneness, dynamic topology and broadcast communication.

Nodes can be distributed strategically or randomly. Therefore, a major challenge found in the WSN research area is related to how to determine the best location for each sensor node taking into account the cost of covering a region, which is considered NP-Hard (Rocha, 2023). Challenges like this can be solved with metaheuristics. A metaheuristic with good possibility of application to this problem is Iterated Local Search (ILS).

The ILS metaheuristic is based on the following idea: apply an initial solution to any initial solution until a local optimum is found, then perturb the found solution and restart the local search. It is important to emphasize that a random restart must be avoided and that the perturbation must be such that it allows escape from the local optimum and allows the exploration of other regions of the search space. The reason for choosing the ILS metaheuristic is linked to what Lourenço, Martin, & Stützle (2019), says about its main characteristics: it is simple, easy to implement, robust and highly effective. Therefore, good results depend mainly on the choice of local search, disturbances and the acceptance criteria. It can be progressively optimized, which means it can be maintained at any level of simplicity. This together with its modular nature of iterated local search gives us the possibility of a short development time and gives ILS an advantage over more complex metaheuristics in the world of industrial applications.

This work presents the proposal of an ILS metaheuristic to determine the best positioning of nodes in a WSN in a given region. It will take into account the reach of the nodes and their cost, and may be a homogeneous or heterogeneous network in relation to the type of node. Next, the performance of the ILS algorithm will be compared with the genetic algorithm presented in Rocha (2023).

The rest of this work is organized as follows. In the second section are presented the description of the problem addressed as well as about WSN. In the third and fourth sections there are respectively the details of the ILS metaheuristic implemented and the computational experiments performed and results obtained. Finally, in the fifth section there are the conclusions about the work developed and future works that can be done.

2. BASIC CONCEPTS USED

Here are shown the basic principles of the concepts that were used to develop this work.

2.1 Wireless Sensor Network

The design of a WSN is influenced by many factors including fault tolerance, scalability, production cost, operating environment, network topology, hardware constraints, transmission medium and power consumption. Each of these factors requires specific requirements in the conception and design of the nodes. The vast majority of wireless sensor nodes are designed in small dimensions, resulting in limitations to the devices in terms of communication, sensing and energy (Akyildiz, Su, Sankarasubramaniam, & Cayirci, 2002).

Despite the limitations of the nodes, work done collaboratively leads to the possibility of carrying out a major task.

In Bouarourou, Zannou, Boulaalam, & Nfaoui (2022), the configuration functionality is considered before sensing, processing and communicating data and it is there that procedures related to planning, positioning and self-organization of a WSN are carried out. The configuration or pre-deployment functionality is related to one or more of the following issues: application requirements, area to be monitored, environment characteristics, choice of nodes, type of WSN and, costs. Certainly, there is great importance in obtaining a good location for positioning each node in the configuration functionality, so that better results can be obtained in the functionalities.

For each application we will have a diversification of characteristics in the configuration functionality. Constraints relating to sensor nodes, the network, the location to be applied, among others, are taken into account. This work only addresses the coverage area, type and cost of the nodes.

2.2 Iterative Local Search Metaheuristic

Lourenço, Martin, & Stützle (2019) talk about the great importance of high-performance algorithms to solve problems that are difficult to optimize. This brings us to metaheuristics. The constant use of metaheuristics in problem solving is due to the non-dependence on knowledge of the problem, as a rough example: metaheuristics find the best solution through exploring the problem, generally in a reasonable period of time.

When being designed, a metaheuristic should preferably follow a standard, which should be: preferably simple, conceptually and in practice; be effective, if possible in its entire structure and; non-dependence on prior knowledge of the problem. Iterated Local Search (ILS) is a metaheuristic method that simplifies meeting all these criteria.

The essence of the ILS metaheuristic can be described in a few words: it iteratively builds a sequence of solutions generated by the built-in heuristic, leading to much better solutions than random attempts to apply this heuristic (Lourenço, Martin, & Stützle, 2019).

An ILS algorithm, as shown in Figure 1, has:

- GenerateInitialSolution (...), where an initial solution s_0 to the problem is generated;
- Local Search(...), search for an improved solution;
- Perturbation(...), modifies the result found, creating an intermediate solution *s´*;

- AcceptanceCriterion (...) Defines at each iteration whether or not the result obtained will be the new local optimum; and
- Stopping criterion (termination condition), determines when the iterative process ends.

Algorithm 1 Iterated Local Search

```
s0 = GenerateInitialSolution
s* = LocalSearch(s0)
repeat
    s' = Perturbation(s*, history)
    s*' = LocalSearch(s')
    s* = AcceptanceCriterion(s*, s*', history)
until termination condition met
```

Figure 1. ILS algorithm pseudocode.

Given a search space *S* of candidate solutions and *s* as a “simple” solution, it is necessary to pay attention to the perturbation applied, as a very small perturbation often leads to falling back into *s** and it will not be possible to explore new solutions. A very large perturbation will make it impossible to maintain the characteristics of the local optimum, with no trend in the result and we will restart as a random algorithm. Figure 2 graphically illustrates the operation of the ILS.

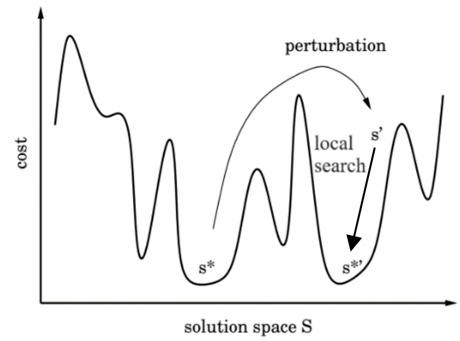


Figure 2. Representation of ILS working. Starting with a local minimum s^*, we apply a perturbation leading to a solution s'. After applying LocalSearch, we find a new local minimum $s^{*'}$ that may be better than s^*. Figure adapted from Lourenço et al. (2019).

2.3 Area Coverage Problem (ACP)

The critical challenge in wireless sensor networks (WSNs) is to strategically position sensors to achieve comprehensive coverage across a designated area. This study introduces a grid-based model for deploying sensor nodes to ensure coverage while maintaining connectivity. By covering relevant points of interest, the proposed model optimizes sensor placement, allowing for an efficient selection of the minimum required sensors, reducing the WSN cost.

The problem in question can be defined as: given an area ***A*** to be monitored, a set of sensor nodes ***S***, a set of demand points ***D***, in this Area Coverage Problem with WSN it must be guaranteed that for each demand point *d* ϵ ***D*** in area *A* at least one *s* ϵ ***S*** cover it. The area coverage problem is illustrated in Figure 3.

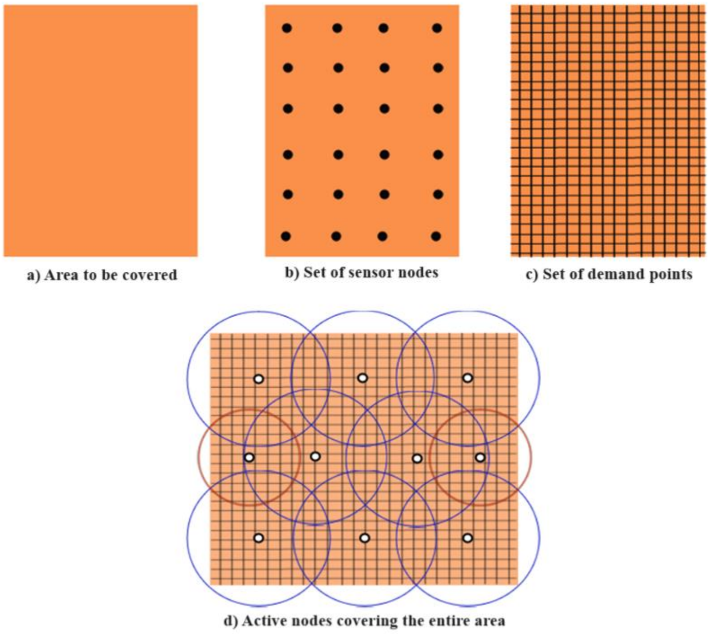


Figure 3. Illustrative example of the area coverage problem for wireless sensor networks. Illustrative example of the area coverage problem for wireless sensor networks.

The area to be mapped within a region is obtained by calculating the rectangle that surrounds it, where the smallest (minlat and minlong) and largest (maxlat and maxlong) values of the latitude and longitude coordinates of the respective area are obtained. The minimum space between two demand points in kilometers (km) is considered to determine the demand points in the area.

We define that the smaller the interval ***i***, the greater the number of demand points d ϵ ***D***, and we will have greater precision in coverage, this being inversely proportional. We have illustrated in Figure 4 the relationship between interval and demand points, where ***A*** = 400 km^2, ***i*** = 2 generating ***D*** = 200. The relationship between the radius at the sensor and the interval of the points of demand must also be taken into consideration. demand. Because the radius of the nodes is smaller than the range of the demand points, it will lead to a solution with little area coverage.

The computational complexity of generating demand points is determined as being $O(mn)=(n + 1)m$, where ***n*** is the number of different sensors, with the most 1 referring to the absence of a sensor and ***m*** is the number of possible points (set of demand points ***D***). This characterizes the problem as having exponential behavior (execution time) depending on the size of the input to guarantee obtaining the optimal solution (Garey and Johnson, 1979), therefore, it is interesting to use heuristics to find good solutions in low computational time.

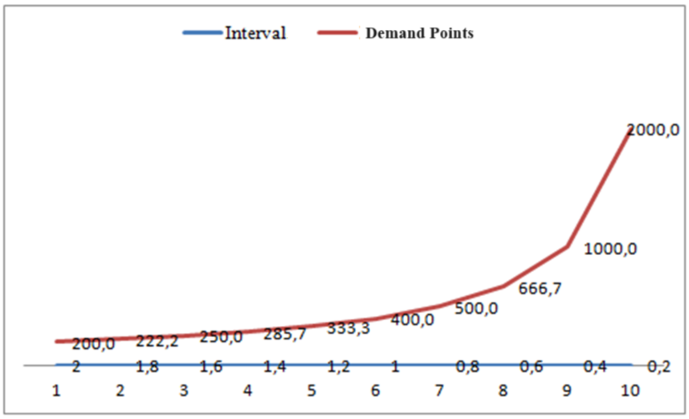


Figure 4. Relationship between interval and demand points.

3. DEVELOPMENT

In this work, an ILS metaheuristic approach is proposed for configuring a WSN to cover a region. The ILS proposed in this work follows the structure of the basic ILS as specified in Lourenço et al. (2019) and presented in Figure 1 of subsection 2.2. The main characteristics of the proposed ILS are presented below.

3.1 Generating Initial Solution

The generated set of demand points ***D*** is represented in an array of size ranging from 0 to ***m*** - 1 with the following rules:

- The array indexes represent each demand point;
- The content of a position in the array is represented by an integer of 0 to ***n*** types of sensor nodes. Where 0 (zero) represents the absence of a sensor node.

To represent the solution structured in the array we have Figure 5, indicating that in d_1=1 (position 1 of the array) a sensor node of type 1 is located, in d_4=4 (position 4 of the array) a sensor of type 2, in position 9 (d_9=9) a type 3 sensor and the other positions do not have any sensor (value 0).

0	1	0	0	2	0	0	0	0	3
0	1	2	3	4	5	6	7	8	9

Figure 5. Representation of the solution in an array.

In this step, the initial solution $\boldsymbol{s}_0$ is generated, which will be our starting point for BuscaLocal $\boldsymbol{s}^*(\boldsymbol{s}_0)$. Obtained ***D***, an array with size $\boldsymbol{m} = |\boldsymbol{D}|$, is populated with 0 (zero = no sensor node) to ***n*** types of sensors randomly ($\boldsymbol{d}_i = k \in \{0,..., \boldsymbol{n}\}$). After checking the cost of the solution, the initial solution is stored for future checks.

3.2 Solution Cost (Acceptance Criteria)

Each solution generated will be subjected to verification of its cost ***C***(***s***). This method scans the array **D**. If di is empty and not covered by the radius of a node, +1 is added to the total cost ***c***, when finding $\boldsymbol{d}_i$ with a node, the cost of the sensor node is added to ***c***. This will be our acceptance criteria for a new solution.

To check whether a demand point $\boldsymbol{d}_i$ is covered or not by a sensor node, the distance between one point and another is calculated as follows:

- At this point our latitudes (lat) and longitudes (long) are already in decimal format.
- The change in lat and long is calculated by subtracting the origin with the destination.
- This change (mud) is converted into radians r=mud*(π/180), remembering that these values cannot be negative.
- We will obtain Δlat, Δlong.
- The results are applied to the following formula: $a = (\sin^2(\Delta lat/2) + \cos(lat1)) * \cos(lat2) * \sin^2(\Delta long/2)$.
- Immediately afterwards the result is applied to an intermediate solution: $c=2*\cot(\sqrt{a}/\sqrt{(1-a)})$, where cot is the inverse of the tangent function, indicated as "tan^−1" on some calculators.
- To obtain the distance in kilometers, the formula d=R*c is used, where R=6.371 km represents the radius of planet Earth.
- Finally, a comparison is made between the radius of the sensor node and the result of the distance calculation.

3.3 Perturbation

At this point we must take into account that if the perturbation applied to the solution is very small it is quite likely that it will remain in the solution to be perturbed, and if it is very large, we can characterize the algorithm as having random restart.

To help the local search to leave a local optimum, perturbation is applied. In array ***D*** where $\boldsymbol{d}_i$=0, a disturbance probability is estimated to add a node of 1 to ***n*** types. Soon after the insertions, the new solution ***s***' is generated, the local search is applied and analyzed by the acceptance criteria. If passed, a new solution ***s**** is generated; if not accepted, the previous ***s**** is returned.

3.4 Local Search

When an initial solution s_0 is generated, the local search is applied. Then the loop begins until the stopping condition where the local search is applied after the perturbation. This technique aims to take the solution from a local optimum to a global optimum.

The local search algorithm was developed as follows. Using the cost of the previous solution (history) as a reference, a loop of size ***m*** was created. When a node is found at $d_i=0$, it is removed and the following process is adopted:

- The cost of the new solution is calculated.
- If the new solution is better, the process in array ***D*** continues from there.
- If the solution is not better, the node is reinserted into d_i and the process continues.

4. COMPUTATIONAL EXPERIMENTS, RESULTS AND DISCUSSION

The results obtained were obtained after comparative execution between the Genetic Algorithm (GA) code of Rocha (2023) and Iterated Local Search on a computer with an Intel Core i3 M370, 2.4GHz, with 4GB of DDR3 1333MHz RAM and Windows 7 Professional operating system. Despite the existence of 2 cores, the algorithm does not exploit this multiprocessing capacity. For ILS was considered as termination condition an execution of 100 times of the main loop as stated in Figure 1.

In 4 problems (with 17, 48, 63 and 78 demand points), 1 instance of each was considered and in the 5th problem 3 instances of the same problem were considered (with 401, 500 and 629 demand points).

First instance of the problem with ***m***=17 we have: 2 types of sensors and range of demand points equal to 0.1 Km2, their costs and radii are in Table 1.

Table 1. First Instance.

	Cost	Radius
Sensor 1	2	0,11
Sensor 2	3	0,13

Second instance of the problem with ***m***=48 we have: 3 types of sensors and range of demand points equal to 0.15 Km^2, their costs and radii are in Table 2.

Table 2. Second Instance.

	Cost	Radius
Sensor 1	2	0,3
Sensor 2	3	0,26
Sensor 3	3	0,4

Third instance of the problem with ***m***=63 we have: 2 types of sensors and range of demand points equal to 0.1 Km2, their costs and radii are in Table 3.

Table 3. Third Instance.

	Cost	Radius
Sensor 1	2	0,11
Sensor 2	3	0,13

Fourth instance of the problem with ***m***=78 we have: 5 types of sensors and range of demand points equal to 0.12 Km2, their costs and radii are in Table 4.

Table 4. Fourth Instance.

	Cost	Radius
Sensor 1	2	0,13
Sensor 2	3	0,19
Sensor 3	4	0,165
Sensor 4	5	0,3
Sensor 5	8	0,2

In the fifth problem, 3 instances were created with the intervals of demand points and number of demand points as follows in Table 5. Their costs and radius are in Table 6.

Table 5. Interval and Total Demand Points.

Interval	Demand Points
0,05	401
0,045	500
0,04	629

Table 6. Fifth Instance.

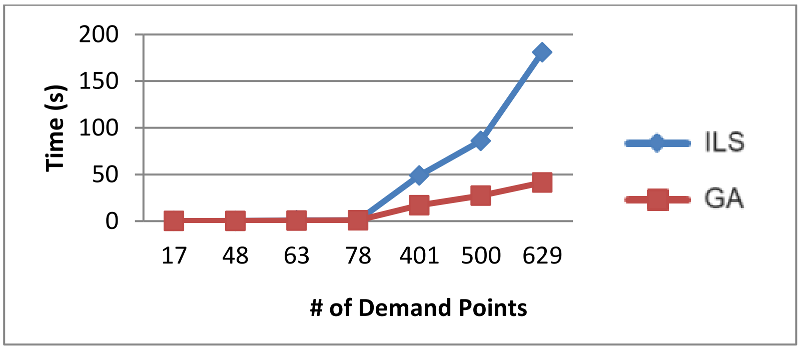
	Cost	Radius
Sensor 1	2	0,11
Sensor 2	3	0,15
Sensor 3	4	0,185
Sensor 4	5	0,095

In Table 7 we have the comparative computational results between the GA and ILS algorithms, considering the instances stated in Tables 1 to 6 and presenting the execution time in seconds and the solution cost.

Table 7. Comparative computational results between GA and ILS.

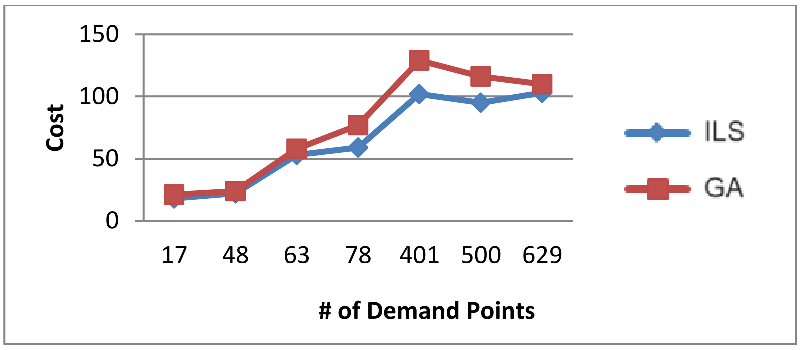
			GA			ILS		
Problem (P)	Instance (I)	# of Demand Point	# of sensors	Time(s)	Cost	# of sensors	Time (s)	Cost
1	1	17	8	0,062	21	7	0,081	18
2	2	48	8	0,221	24	7	0,242	22
3	3	63	21	0,423	58	18	0,751	53
4	4	78	18	0,801	77	14	0,792	59
5	5	401	37	16,971	129	29	49,022	102
	6	500	35	27,221	116	25	85,863	95
	7	629	32	41,341	110	28	180,962	103

Related to execution time versus the number of demand points, comparing GA with ILS algorithm as show in Graphic 1, ILS is more efficiently only in the fourth instance. In the fifth problem we can obtain an increasingly unfavorable result. At ***m***=401 the execution time was 65.38% higher, at ***m***=500 it was 68.30% and at ***m***=629 it was 77.15% when comparing GA with ILS.

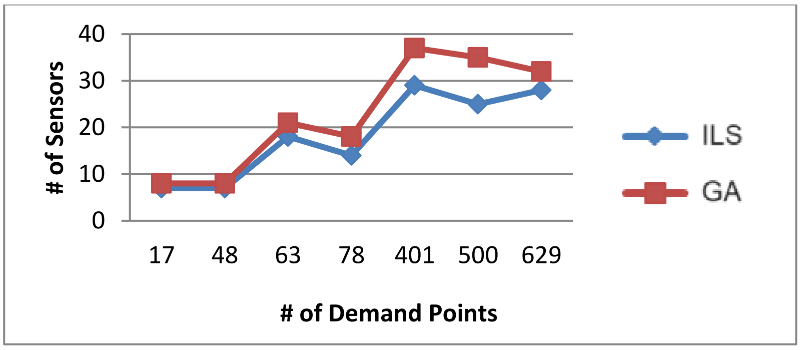


Graphic 1. # of Demand Points vs Time.

Comparing the results of the ILS algorithm versus GA, ILS obtained better local optima in all instances, being on average 14.28% better, as shown in Graphic 2 and 17.43% better related to the number of sensor nodes used, as presented in Graphic 3.



Graphic 2. # of Demand Points vs Solution Cost.



Graphic 3. # of Demand Points vs # of Sensor Nodes.

5. CONCLUSIONS

This work addresses the Area Coverage Problem (ACP) and proposes an algorithm called Iterated Local Search to obtain better results than the Genetic algorithm proposed in Rocha (2023). This work focuses on the problem of area coverage, thus making it a viable solution to other types of problems outside the scope of Wireless Sensor Networks. We can cite as an example the best positioning of crop irrigation pivots and radar positioning.

The local search method followed by perturbation applied so as not to generate a random solution proved to be satisfactory. For all solutions, the ILS algorithm determined a better cost solution with a smaller number of sensor nodes. However, in large problems the computational time showed exponential behavior.

The results obtained by the Iterated Local Search (ILS) Algorithm proposed for PCA are satisfactory when compared to the main work of Rocha (2023). However, larger studies and new developments can be carried out in order to seek efficiency in the ILS proposed in this work. As suggestions, there are:

- Restructuring of the parameters passed to the local search to obtain better results in acceptable computational times.
- Carry out computational tests with instances of even greater dimensionality.

- Include the possibility of more than 1 type of sensor node in a region.
- All ILS computational tests were carried out with 100 repetitions. If the bottleneck that increases computational time is removed, the number of repetitions can be increased, reaching better solutions.

ACKNOWLEDGEMENTS

The authors acknowledge the financial support of the Fundação de Amparo à Pesquisa do Tocantins - FAPT/Governo do Tocantins, granted through research productivity grant nº 01/2019.

REFERENCES

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. ***IEEE Communications magazine****, 40*, 102–114.

BenSaleh, M. S., Saida, R., Kacem, Y. H., & Abid, M. (2020). Wireless sensor network design methodologies: A survey. ***Journal of Sensors****, 2020*, 9592836.

Bouarourou, S., Zannou, A., Boulaalam, A., & Nfaoui, E. H. (2022). S**ensors Deployment in IoT Environment. *International Conference on Digital Technologies and Applications*,** (pp. 276–283).

Elson, J., & Estrin, D. (2004). ***Sensor networks: a bridge to the physical world.* Springer**.

Estrin, D., Girod, L., Pottie, G., & Srivastava, M. (2001). I**nstrumenting the world with wireless sensor networks. *2001*** *IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, *4*, pp. 2033–2036.

Gartner. (2023). Gartner Forecasts Worldwide Semiconductor Revenue to Grow 17% in 2024. *Gartner Forecasts Worldwide Semiconductor Revenue to Grow 17% in 2024*.

Gary, M. R., & Johnson, D. S. (1979). C**omputers and Intractability: A Guide to the Theory of NP-completeness.** WH Freeman and Company, New York.

Lourenço, H. R., Martin, O. C., & Stützle, T. (2019). Iterated local search: Framework and applications. ***Handbook of metaheuristics*,** 129–168.

Rocha, M. L. (2023). A Genetic Approach Applied to the Setup of a Wireless Sensors Network to Monitoring a Location. ***Revista Cereu****s, 15*, 289–298.

Tilak, S., Abu-Ghazaleh, N. B., & Heinzelman, W. (2002). Infrastructure tradeoffs for sensor networks. *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, (pp. 49–58).

Yingshu, L., My, T., & Weili, W. (2008). Wireless sensor networks and applications. ***Springer series on signals and communication technology***.