Optimizing Wireless Sensor Network Planning: Integrating Biased Random-Key Genetic Algorithm and Local Branching for Scalable Solutions

Almir Antônio Monteiro Junior
System Engineering and Computer Science Program
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
amonteirojr@cos.ufrj.br

Diego Brandão
Computer Science Postgraduate Program
Federal Center for Technological Education of Rio de Janeiro
Rio de Janeiro, Brazil
diego.brandao@cefet-rj.br

Felipe da Rocha Henriques
Computer Science Postgraduate Program
Federal Center for Technological Education of Rio de Janeiro
Rio de Janeiro, Brazil
felipe.henriques@cefet-rj.br

Claudio M. de Farias
System Engineering and Computer Science Program
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
cmicelifarias@cos.ufrj.br

Pedro Henrique González
System Engineering and Computer Science Program
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
pegonzalez@cos.ufrj.br

***Abstract*—This study addresses the Wireless Sensor Network Planning Problem with Multiple Sources/Destinations (WSNP-MSD), an optimization challenge focused on reducing the sensor count within a network topology for a specified area, considering numerous sources and destinations. We introduce a hybrid strategy for tackling WSNP-MSD, particularly effective for large-scale scenarios, combining a Biased Random-key Genetic Algorithm with a Local Branching Technique. This methodology is justified by the limitations exact methods may encounter when the number of variables increases. Through computational experiments, we demonstrate the superiority of our proposed method over conventional exact methods in managing large instances of the WSNP-MSD.**

***Index Terms*—Wireless Sensor Network, Genetic Algorithm, BRKGA, Local Branching, Graph, Topology.**

I. Introduction

We can define Wireless Sensor Networks (WSNs) as spatially distributed sensors and one or more edge devices deployed in a region of interest, to perform some kind of monitoring [1]. These sensors can execute several tasks, like collecting and processing data, aside from communication, using some type of wireless network standard [2], [3]. This type of network can be applied to countless real-world applications, encompassing environmental signal monitoring [4], military applications [5], and the monitoring of biometric data [6].

Lately, WSNs have been utilized as infrastructure for the Internet of Things (IoT) [7], [8]. In this application, sensors transmit the data to one or more destinations [9]. The device receiving traffic is known as an edge device, commonly having unlimited energy capacity, while the other sensors have limited energy and processing capabilities. Given that communication is the task that consumes most of the energy in this kind of application [4], data processing is a commonly explored solution in these networks [10], [11].

Another aspect of this kind of network is the network topology control [12]. In this context, the topology is designed to minimize the number of sensor nodes needed to cover the area of interest [13]. To the best of our knowledge, in scenarios where the objective is to minimize the number of sensors in the topology while accommodating multiple source and destination sensor nodes, only [14] introduced a method, whose results are compared to the method proposed by this research.

This work investigates an approach to solve the Wireless Sensor Network Planning Problem with Multiple Sources/Destinations (WSNP-MSD), first presented by [14], using a hybrid method composed of a Biased Random-key Genetic Algorithm (BRKGA) and a Local Branching Technique. The WSNP-MSD aims to minimize the number of sensors required to deploy a network consisting of multiple sources (sensors) and multiple destinations (edge devices), while ensuring communication between all the source/destination pairs. This hybrid approach is justified by the fact that solving large instances of the problem can become impractical due to several reasons, like running time or memory issues. Therefore, the proposed method starts using the metaheuristic component, being able to handle the analysis of huge instances, and then refine the results, by applying the Local Branching procedure in the solution previously found.

To validate the effectiveness of the proposed methods, computational experiments were conducted on randomly generated instances. The obtained results indicate that the proposed approach is indeed more adept at managing larger instances, in comparison to the exact method, effectively reducing the number of sensors deployed in the constructed network.

The structure of this paper is organized as follows: Section II introduces and provides a detailed explanation of the problem under investigation. Section III outlines the methodology employed in this study. Section IV elaborates on the conducted experiments and discusses their outcomes. Finally, Section V summarizes the conclusions and suggests possible topics for future research.

II. Problem Definition

This section introduces the Wireless Sensor Network Planning Problem with Multiple Sources/Destinations (WSNP-MSD). Presented by [14], WSNP-MSD states that, given an area that is intended to be covered by sensors, one must find the best topology of this network, ensuring that all communication will flow from the defined sources to the destination sensors, while using as few as possible intermediate sensors, to reduce the costs.

To construct the network topology, the problem formulation encompasses two primary sets: a set S of sensors, where each sensor $s \in S$ is associated with a set of potential positions s_i for $i = 1..k_s$, an allocation cost c_s, and a communication radius r_s; and a set P of pairs of sources and destinations $p = (o_p, d_p)$. It is required that for all pairs $p = (o_p, d_p) \in P$, considering the communication radii of the sensors, each source o_p must be capable of communicating with its corresponding destination d_p, either directly or via multiple relay hops [14].

Before the creation of the mathematical model, the authors initiated the process by constructing an auxiliary database utilizing an artificial graph $G = (V, E)$. To compile the set V, two steps were taken: initially, for each potential position s_i of a sensor $s \in S$, a vertex v_{s_i} was generated and incorporated into a set $\bar{V}$. Subsequently, for each sensor s identified within the list of source/destination pairs, an artificial node was generated and appended to a set A. Consequently, the set V was established as $V = \bar{V} \cup A$. To define the set E, two additional actions were executed: firstly, for each pair of vertices $v_{s_i}, v_{\bar{s}i} \in \bar{V}$, where $s \neq \bar{s}$ and their communication radii intersect, an edge was created and added to E. Secondly, for every vertex $a_s \in A$ and $vs_i \in \bar{V}$, where s_i represents a feasible location for s, an edge (a_s, v_{s_i}) was established and included in E. Subsequently, the original sources and destinations were substituted with the artificial vertices representing the respective source and destination nodes. Leveraging the constraints previously outlined and the auxiliary database constructed, an integer programming formulation that represents the WSNP-MSD can be defined as:

$$\min \quad \sum_{i \in V} c_i y_i$$

$$\text{s.a.} \sum_{(i,j)\in\delta^+(i)} x_{ij}^p - \sum_{(j,i)\in\delta^-(i)} x_{ji}^p = b_i^p, \qquad \forall i \in V,\, p \in P \quad (1)$$

$$\sum_{\alpha \in C(i)} y_\alpha \leq 1, \qquad \forall i \in S \quad (2)$$

$$x_{ij}^p \leq y_i, \qquad \forall (i,j) \in E, p \in P \quad (3)$$

$$x_{ij}^p \leq y_j, \qquad \forall (i,j) \in E, p \in P \quad (4)$$

$$y_i = 1, \qquad \forall i \in A \quad (5)$$

$$x_{ij}^p \in \{0,1\}, \qquad \forall (i,j) \in E, p \in P \quad (6)$$

$$y_i \in \{0,1\}, \qquad \forall i \in V \quad (7)$$

where,

$$b_i^p = \begin{cases} -1 & \text{if } i = d_p, \\ 1 & \text{if } i = o_p, \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

The family of Constraints (1) guarantees the existence of a communication pathway between every pair of source and destination. The family of Constraints (2) ensures that each sensor is deployed no more than once, occupying only a single potential position. This is achieved through the use of an auxiliary structure $C(i)$, which denotes the set of all potential positions for a given sensor i. Constraints (3) and (4) stipulate that communication between vertices i and j is possible only if both vertices are allocated. Moreover, the family of Constraints (5) mandates that all auxiliary vertices associated with each source/destination pair, and thereby included in set A, must be accounted for in the solution. Lastly, Constraints (6) and (7) state the domain for the decision variables.

III. Methodology

This section outlines the methodology introduced in this study. The approach consists of two key components: the Biased Random-Key Genetic Algorithm (BRKGA), responsible for exploring the solution space using genetic components, and the Local Branching technique, employed to refine the quality of the best solution found by BRKGA. Detailed explanations of each component are provided in the subsequent subsections.

1) Biased Random-Key Genetic Algorithm (BRKGA): Genetic algorithms draw significant inspiration from Darwin's theory of evolution, which posits that the most adaptable individuals are more likely to survive [15]. Starting with an initial group, these algorithms identify the most promising solutions, referred to as chromosomes, using a measure known as the Fitness Function to evaluate their effectiveness. The superior solutions are then used to produce offspring, inheriting traits from their predecessors, thereby enhancing the likelihood of yielding the most advantageous solutions in subsequent generations [16]. This iterative process continues until a predefined termination criterion is reached, such as reaching a maximum duration, the attainment of a certain number of generations, or a stagnation in generational improvement.

Within the spectrum of genetic algorithms, the Biased Random-Key Genetic Algorithm (BRKGA) [17] represents an advanced iteration of the Random-Key Genetic Algorithm (RKGA) [18]. RKGA was pivotal for introducing two essential concepts to genetic algorithms: the encoder, which maps a potential solution to a continuum between 0 and 1, and the decoder, which reverts these encoded values to their original form. BRKGA builds upon RKGA by incorporating a preferential selection mechanism favoring the elite members of the current generation. This preference enhances their likelihood of transmitting their genetic material to the next generation during the crossover process.

In this work, each chromosome is a vector with n positions, where n is the number of possible positions of sensors in the instance. Being the values of each decision variable regarding the position in the domain $[0, 1]$, when the position value was higher than 0.5, it was marked as used, and marked as not used otherwise. However, this approach could lead to an invalid solution, because the same sensor can not be in two places at the same time. To overcome this, for every group of possible positions linked to a sensor, the one with the highest value was chosen. When there were no positions with values higher than 0.5 inside a group, the sensor was considered as not used, excluding the ones in the source/destination pairs, where the position with the highest value was chosen, regardless of whether it was over 0.5 or not. This is explained by the fact that every sensor in the source/destination list must also be in the solution, rendering it infeasible otherwise.

After this encoding/decoding step, a viability check is run to ensure that the chromosome is viable. For each source/destination pair, the algorithm tries to find the minimum path between them. If it is inexistent for at least one of the pairs, the chromosome is marked as infeasible, with its fitness value set to $+\infty$. Otherwise, the chromosome receives as its fitness value the number of sensors used. The whole process is detailed in Figure 1.

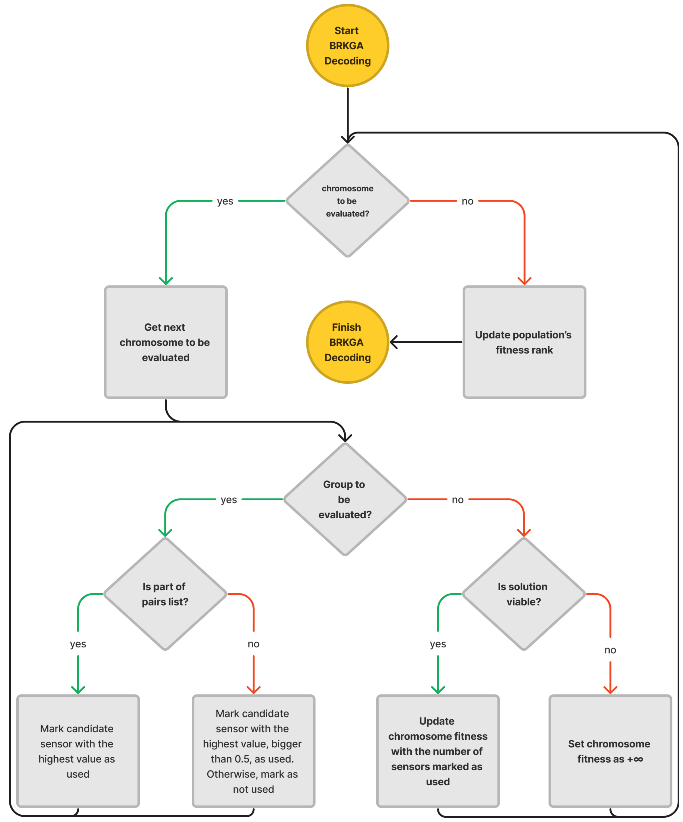

Fig. 1: Decoding procedure flow

A. *Local Branching*

The Local Branching (LB) technique, introduced by [19], can be used to enhance existing feasible solutions. It is designed for efficient exploration of the search space, aiming to produce superior heuristic solutions by employing a mixed-integer programming (MIP) solver. Since it can be used as a local search strategy, Local Branching integrates linear inequalities into the mathematical model to delineate neighborhoods that are thoroughly explored. Several notable studies have demonstrated the efficacy of this strategy when solving optimization problems [20], [21].

In the context of the problem addressed in this paper, consider a solution $s \in P$, with P representing the polyhedron defined by the constraints from 1 to 5. We adopt this methodology by appending the subsequent LB constraint to our model:

$$\sum_{i \in V | \bar{y}_i = 0} y_i + \sum_{i \in V | \bar{y}_i = 1} (1 - y_i) \leq \Delta \tag{9}$$

where Δ is a positive integer that indicates the number of variables that can have their value swapped from one to zero and vice-versa.

IV. Computational Experiments and Results

This section provides an overview of the conducted experiments and their outcomes. In Section IV-A, we detail the process for generating random instances and the specific ones used to assess the proposed methodology. Following that, Section IV-B examines the components of the proposed approach, evaluating the contribution of each one to the overall proposed method. Finally, a comparison of the results obtained by the proposed method against existing solutions in the literature is presented in Section IV-C.

The experiments were conducted on a machine equipped with an AMD Ryzen™ 5 5600G processor operating at 3.9 GHz and 32GB of RAM. For the implementation of the proposed method, we utilized the Python version developed in [22], along with IBM's CPLEX Mathematical Optimization and the DOCplex Python libraries, all within a Python 3.10 framework. The devised method was applied to each instance ten times, with each run constrained by a 3600-second time limit.

A. *Instances*

To evaluate the methodology proposed in this study, a series of instances of varying sizes was generated since the available instances were of limited size. To do that, initially, a set of random points is distributed within a rectangular space, which are then grouped based on their proximity through the KMeans algorithm [23]. Each grouping represents a potential sensor location, with every point within a cluster signifying a possible deployment position for a sensor. Following this, to establish source/destination pairs, the shortest path connecting each potential sensor location is computed, from which a random selection of feasible pairs is made. For the construction of the distance matrix, the distance between two potential positions is determined using the Manhattan distance metric, but only if their communication ranges overlap. The communication range for each sensor is set to 5 meters and the cost per sensor is standardized to 1, following the parameters specified by the original authors. The generated instances are characterized by three variable factors: the number of potential sensors, the deployment positions available for each sensor, and the count of source/destination pairs. Figure 2 depicts the instance creation process, and the specifics of the generated instances are detailed in Table I.

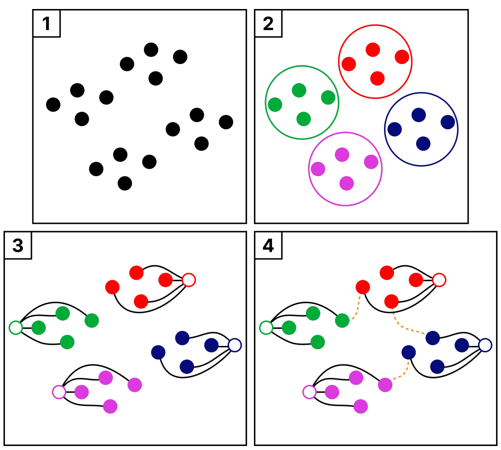

Fig. 2: Steps to generate an instance. In Subfigure 1, a set of random points is generated; in Subfigure 2, they are clustered using KMeans. Every cluster is a sensor, and every point inside a cluster is a possible position to deploy the sensor; in Subfigure 3, artificial nodes are created for every cluster and linked to every possible position of the cluster; in Subfigure 4, based on the sensor's communication radius, when two radii from different cluster elements overlap, an edge between them is generated.

Instance ID	Total Possible Positions	Sensors	Pairs
001	200	20	5
002	200	20	10
003	400	40	10
004	400	40	20
005	600	60	20

TABLE I: Details of the instances generated and analyzed in this work

B. Ablation Study

This subsection is dedicated to demonstrating the impact of the local branching technique on the quality of the solution by comparing outcomes before and after its application. Each trial was allocated 3600 seconds, with any time remaining after the execution of BRKGA being allocated to the local branching procedure. The specific settings for the hyperparameters utilized by both BRKGA and the local branching process are outlined in Table II. Despite the fixed time limit, the termination conditions were established after careful consideration as either reaching 200 generations in total or experiencing 100 consecutive generations without any improvement.

Component	Parameter	Value
BRKGA	Independent Populations	1
	Elite Percentage	30%
	Mutation Percentage	15%
Local Branching	Delta (Δ)	$int(possible\_positions/2)$

TABLE II: Hyperparameters used in the proposed method

For smaller instances, the local branching technique significantly enhanced the solutions generated by BRKGA. In the case of larger instances, while the improvement margin was narrower, enhancements were still observed in the majority of cases. Specifically, instances 001, 002, and 003 exhibited the most notable improvements with the incorporation of local branching, highlighting its pivotal role within the methodology proposed in this paper and underscoring the rationale for its inclusion. Table III provides a comprehensive overview of the results, including execution times and observed gaps, whereas Figure 3 visually represents the progression of the BRKGA component across generations. Except for instance 005, the outcomes achieved with BRKGA alone fell short of those attained by employing the full methodology proposed.

The carried-out experiments demonstrate the significance of each element within the proposed methodology. Although the Biased Random-Key Genetic Algorithm (BRKGA) is capable of identifying solutions within a concise timeframe, the application of the local branching technique further refines these initial findings, resulting in enhancements exceeding 60% in certain instances.

C. Proposed Method vs Literature

Having illustrated the critical role of each component within our approach, this study proceeds to benchmark the performance of our methodology against existing procedures documented in the literature. We applied the integer programming model introduced by [14] to each instance described in Section IV-A. Similar to our experiments, a time constraint of 3600 seconds was imposed. The outcomes, when juxtaposed with those achieved by our proposed method, are detailed in Table IV.

For the smaller instances, specifically 001, 002, and 003, the proposed method obtained the optimum solution, as validated by the integer programming model, albeit requiring more time compared to the approach reported in the literature, which

Instance ID	BRKGA*		BRKGA + Local Branching*		
	Sensors [min, max]	Time (s) [min, max]	Sensors [min, max]	Time (s) [min, max]	Gap [min, max]
001	14.4 [14, 16]	115.9 [98.47, 143.14]	**8** [**8**, **8**]	306.33 [254.57, 343.2]	**-57.14%** [**-54.54%**, **-66.67%**]
002	15.8 [15, 16]	122.32 [98.24, 162.88]	**12** [**12**, **12**]	801.04 [751.18, 915.54]	**-27.34%** [**-22.22%**, **-28.57%**]
003	24.9 [23, 27]	519.85 [506.6, 530.43]	**15** [**15**, **15**]	1923.69 [1704.62, 2402.1]	**-49.62%** [**-42.1%**, **-57.14%**]
004	30.6 [28, 33]	638.6 [486.88, 694.17]	**29.7** [**28**, **30**]	3543.56 [2467.53, 3600]	**-2.98%** [**-8.87%**, **-1.98%**]
005	42.9 [41, 44]	1949.35 [1827.89, 2026.04]	42.9 [41, 44]	3600 [3600, 3600]	0% [0%, 0%]

* Mean of 10 executions

TABLE III: Results of the proposed method with and without the local branching component

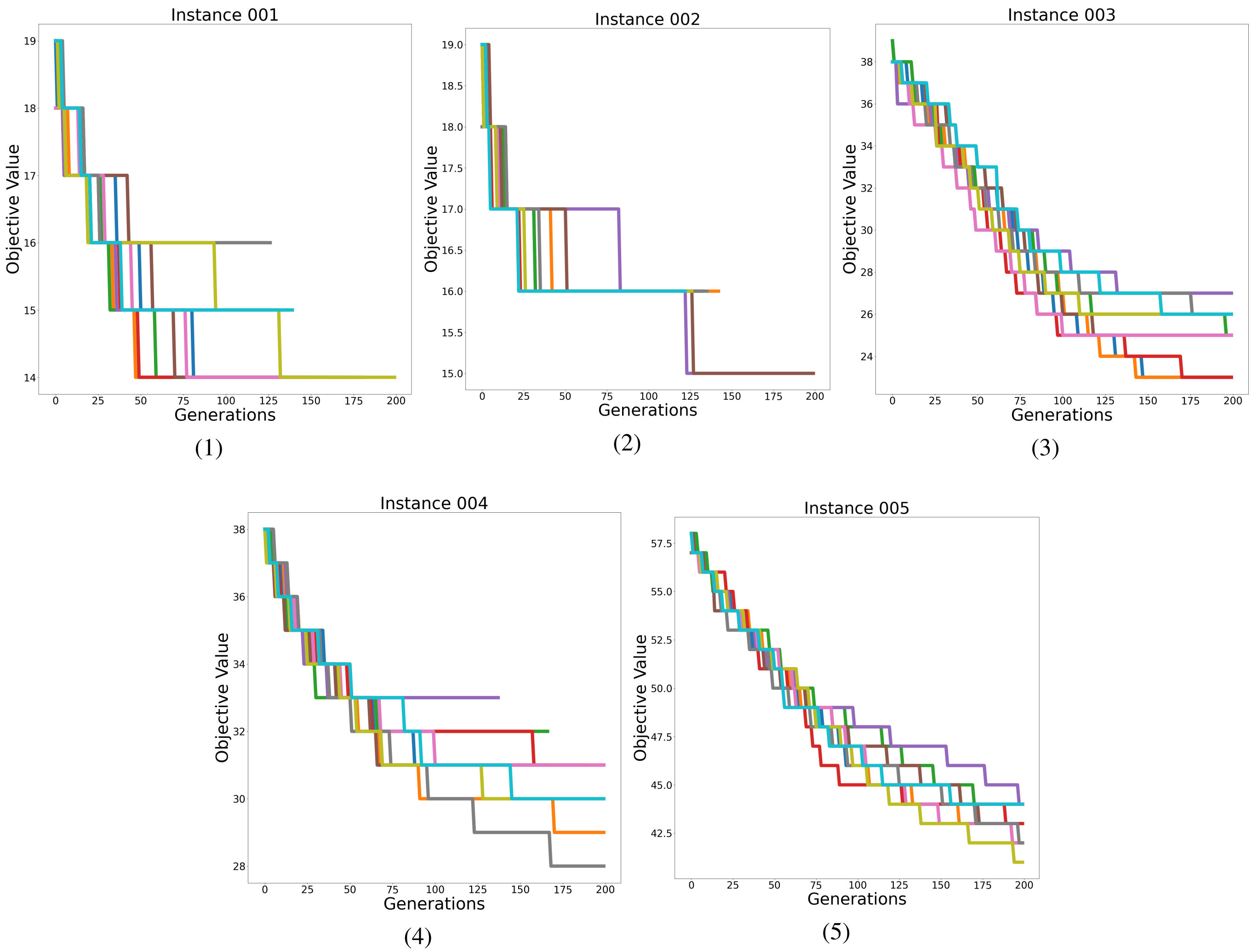

Fig. 3: Evolution of the result of each run of BRKGA, showing a constant decrease in the result as the number of generations increases, but not reaching the results achieved by local branching

is expected. Conversely, for the larger instances, our method either surpassed or matched the performance of the strategies documented in the literature, achieving these results within the same or a shorter timeframe.

V. Conclusions and Future Works

This paper proposes a new methodology for addressing the Wireless Sensor Network Planning Problem with Multiple Sources/Destinations (WSNP-MSD). The methodology comprises two distinct components: a modified genetic algorithm known as the Biased Random-Key Genetic Algorithm (BRKGA) and a Local Branching procedure that imparts deterministic characteristics to the latter stage of the method, aiming to enhance the best solution found by BRKGA.

First, to demonstrate the impact of the local branching on the quality of the results, the values before and after local branching were compared. The experiments demonstrated that adding this deterministic component to refine the solution achieved by BRKGA improved the final solution by more than 60%. Next, the proposed method was compared to the current literature. The method proposed by this work achieved similar or better results if compared to the integer programming model

Instance ID	Literature			BRKGA + Local Branching*		
	Sensors	Time (s)	Gap	Sensors [min, max]	Time (s) [min, max]	Gap [min, max]
001	8	7.67	Optimal	**8** [**8**, **8**]	306.33 [254.57, 343.2]	Optimal
002	12	17.59	Optimal	**12** [**12**, **12**]	801.04 [751.18, 915.54]	Optimal
003	15	490.52	Optimal	**15** [**15**, **15**]	1923.69 [1704.62, 2402.1]	Optimal
004	30	3600	5.26%	**29.7** [**28**, **30**]	3543.56 [2467.53, 3600]	**-1%** [**-6.9%**, 0%]
005	42	3600	16.67%	42.9 [**41**, 44]	3600 [3600, 3600]	2.12% [**-2.41%**, 4.65%]

* Mean of 10 executions

TABLE IV: Results of the proposed method compared to the method found in the literature

present in the literature, taking less or the same time. The improvements were up to 6.9%.

As expected, the integer programming model was the best fit when evaluating small to medium-sized instances. However, in large instances, the proposed approach was able to improve or at least reach similar results if compared to the ones achieved by the integer programming model. Therefore, the method proposed in this paper achieved the expected result, finding valuable results when solving the WSNP-MSD, in instances with a large number of candidate positions to deploy the sensors.

For future works, an interesting approach would be the usage of other hybrid methods, focusing on reducing the execution time. For example, Reinforcement Learning could be used to filter promising chromosomes, reducing the time spent checking lower-quality individuals of each generation.

ACKNOWLEDGEMENT

This research was partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq and Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro - FAPERJ.

REFERENCES

[1] M. R. Senouci and A. Mellouk, "1 - wireless sensor networks," in *Deploying Wireless Sensor Networks*, M. R. Senouci and A. Mellouk, Eds. Elsevier, 2016, pp. 1–19. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9781785480997500015

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, March 2002.

[3] N. Temene, C. Sergiou, C. Georgiou, and V. Vassiliou, "A survey on mobility in wireless sensor networks," *Ad Hoc Networks*, vol. 125, p. 102726, 2022.

[4] F. R. Henriques, L. Lovisolo, and M. G. Rubinstein, "DECA: distributed energy conservation algorithm for process reconstruction with bounded relative error in wireless sensor networks," *EURASIP Journal on Wireless Com. and Net.*, vol. 2016, no. 163, pp. 1–18, July 2016.

[5] T. Azzabi, H. Farhat, and N. Sahli, "A survey on wireless sensor networks security issues and military specificities," in *2017 International Conference on Advanced Systems and Electric Technologies (IC˙ASET)*, 2017, pp. 66–72.

[6] C. B. de M. Ferreira, R. M. Guedes, and F. d. R. Henriques, "Cluster-head switching algorithm based on node temperature in wireless body sensors networks," in *WebMedia 2019*, ser. WebMedia 2019, 2019, p. 65–72.

[7] A. Al-Fukaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, *Internet of Things: A Survey on Enabling Technologies, Protocols and Applicationa*. IEEE, 2015, pp. 2347–2376.

[8] S. Sharma and V. K. Verma, "An integrated exploration on internet of things and wireless sensor networks," *Wireless Personal Communications*, vol. 124, no. 3, pp. 2735–2770, 2022.

[9] S. He, K. Shi, C. Liu, B. Guo, J. Chen, and Z. Shi, "Collaborative sensing in internet of things: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 24, no. 3, pp. 1435–1474, 2022.

[10] F. Da Rocha Henriques, L. Lovisolo, and E. A. Barros da Silva, "Rate-distortion performance and incremental transmission scheme of compressive sensed measurements in wireless sensor networks," *Sensors*, vol. 19, no. 2, 2019.

[11] F. d. R. Henriques, L. Lovisolo, and E. A. B. da Silva, "On the design of klt-based maximally incoherent deterministic sensing matrices for compressive sensing applied to wireless sensor networks data," *Wireless Networks*, pp. 1–14, 2023.

[12] M. Khalily-Dermany, M. Shamsi, and M. Nadjafi-Arani, "A convex optimization model for topology control in network-coding-based-wireless-sensor networks," *Ad Hoc Networks*, vol. 59, pp. 1 – 11, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870516303389

[13] J. A. de Brito, D. R. Totte, F. O. Silva, J. R. d. P. Junior, F. d. R. Henriques, L. Tarrataca, D. B. Haddad, and L. S. de Assis, "Memetic algorithm applied to topology control optimization of a wireless sensor network," *Wireless Networks*, vol. 28, no. 8, pp. 3677–3697, 2022.

[14] A. A. Monteiro Junior, L. Faulhaber, F. d. R. Henriques, and P. H. González, "Optimizing the topology of wireless sensor networks with multiple sources and destinations," *The Latin American Workshop on Information Fusion*, 2023.

[15] C. Darwin and D. Knight, *The Evolution Debate, 1813-1870: On the Origin of Species*, ser. Evolution debate, 1813-1870. Routledge, 2003. [Online]. Available: https://books.google.com.br/books?id=00AOjlmLhS4C

[16] A. N. Sloss and S. Gustafson, "2019 evolutionary algorithms review," *Genetic programming Theory and practice XVII*, pp. 307–344, 2020.

[17] J. F. Gonçalves and M. G. C. Resende, "Biased random-key genetic algorithms for combinatorial optimization," *Journal of Heuristics*, vol. 17, pp. 487–525, 2011.

[18] J. C. Bean, "Genetic algorithms and random keys for sequencing and optimization," *ORSA Journal on Computing*, vol. 6, pp. 154–160, 1994.

[19] M. Fischetti and A. Lodi, "Local branching," *Mathematical Programming*, vol. 98, pp. 23–47, 2003.

[20] P. H. Gonzalez and J. Brandão, "A biased random key genetic algorithm to solve the transmission expansion planning problem with re-design," in *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2018, pp. 1–7.

[21] P. H. González, L. Simonetti, P. Michelon, C. Martinhon, and E. Santos, "A variable fixing heuristic with local branching for the fixed charge uncapacitated network design problem with user-optimal flow," *Computers Operations Research*, vol. 76, pp. 134–146, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054816301484

[22] C. E. Andrade, R. F. Toso, J. F. Gonçalves, and M. G. Resende, "The multi-parent biased random-key genetic algorithm with implicit path-relinking and its real-world applications," *European Journal of Operational Research*, vol. 289, no. 1, pp. 17–30, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221719309488

[23] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," *Information Sciences*, vol. 622, pp. 178–210, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025522014633