A Novel Approach to Enhance LoRaWAN Performances Based on Optimization Algorithms

Yassine Latreche[a,b], Mokhtar Essaid[b], Mahmoud Golabi[b], Ismail Bennis[b], and Lhassane Idoumghar[b]
[a]Higher School in Computer Science,May 8[th], 1945,22016 Sidi Bel Abbés, Algeria
[b]IRIMAS UR 7499, University of Haute-Alsace, 2 Rue des Frères Lumière, Mulhouse, France
ya.latreche@esi-sba.dz, [mokhtar.essaid, mahmoud.golabi, ismail.bennis, lhassane.idoumghar]@uha.fr

***Abstract*—Thanks to its long-range coverage and low-power characteristics, LoRaWAN (Long Range Wide Area Network) technology has emerged as a prominent communication platform for Internet of Things (IoT) applications. However, enhancing the performance of LoRaWAN networks, especially in environments marked by large packet sizes, frequent packet transmissions, or extensive device deployments, continues to pose a significant challenge. This paper introduces an approach to enhance the packet delivery ratio (PDR) in LoRaWAN networks using optimization algorithms for dynamic Spreading Factor (SF) selection. We propose an alternative to the Adaptive Data Rate (ADR) mechanism in LoRaWAN by leveraging three well-established optimization algorithms: Genetic Algorithm (GA), Simulated Annealing (SA), and Differential Evolution (DE). Each algorithm is applied to the SF selection problem to maximize the PDR. The results show that the SA algorithm is the most promising in enhancing network performance. By maximizing PDR, the proposed approach aims to improve the reliability and efficiency of communication in LoRaWAN deployments, thereby enabling more robust and resilient IoT applications.**

***Index Terms*—LoRaWAN, Optimization Algorithms, ADR.**

I. INTRODUCTION

The emergence of Internet of Things (IoT) applications in various fields has led to the widespread use of LoRaWAN (Long Range Wide Area Network) technology, which provides long-range connectivity and low-power operation for sensor and actuator networks to connect to the internet [1]. In LoRaWAN, selecting the Spreading Factor (SF) is key to balancing communication range, data rate, and energy consumption. Higher SF values provide longer range but lower data rates and greater energy use [2]. Conversely, lower SF values yield faster data rates but reduce range and increase susceptibility to interference [2].

As shown in [3], choosing the right SF is extremely important for maximizing the Packet Delivery Ratio (PDR), which is the ratio of successfully received packets to the total transmitted packets. Maximizing PDR is essential for ensuring reliable communication in LoRaWAN networks, especially in situations where there are large packet sizes, frequent packet transmissions, or a large number of devices deployed.

Thus, selecting an appropriate SF is critical in balancing the trade-offs between communication range, data rate, and energy consumption in LoRaWAN networks [4]. When SF values are incorrectly set, several challenges arise like, sub-optimal network performance, exhibiting a poor PDR, increasing latency, and decreasing overall efficiency. Also, selecting the wrong SF can exacerbate interference and congestion problems in the network. While higher SF values narrow the signal bandwidth and reduce interference, they can also limit network capacity and increase latency. Conversely, lower SF values increase susceptibility to interference, leading to packet collisions and decreased reliability [5].

One of the fundamental features of LoRaWAN designed to optimize network performance is ADR (Adaptive Data Rate) [6]. It dynamically adjusts communication parameters, such as SF, based on real-time environmental conditions and device requirements. ADR operates at the network server level and communicates with end devices to optimize their transmission parameters. It continuously monitors various metrics such as signal strength, received signal quality, and packet error rate to assess the current state of the network and environmental conditions. In addition, ADR utilizes link budget calculations to estimate the optimal SF for each device. The link budget considers factors, such as distance to the gateway, path loss, interference level, and device sensitivity to determine the appropriate SF for maximizing communication range while maintaining reliable data transmission [7]. In accordance with the calculated link budget and predefined network constraints, ADR dynamically adjusts the SF for end devices in order to optimize network performance. Devices experiencing favorable conditions may be assigned lower SF values to increase data rates, while devices facing interference or communicating over long distances may be assigned higher SF values to enhance signal robustness and range.

However, while ADR offers significant benefits regarding network performance, some limitations emerge, particularly in scenarios with a large number of devices or rapidly changing environmental conditions. The following are the main limitations that ADR faces [8]:

- **Scalability:** ADR becomes less effective in large-scale LoRaWAN networks, where growing device numbers compete for network access, leading to higher latency and reduced performance.
- **Adaptation Time:** ADR's slow adaptation to changing conditions due to the number of needed messages to make decisions [9], which can impact responsiveness and reduce throughput in real-time applications with high interference.

The main goal of this research is to discover a substitute for the ADR mechanism in LoRaWAN networks by using optimization algorithms like Genetic Algorithm (GA), Simulated Annealing (SA), and Differential Evolution (DE). Our aim is to enhance SF selection and scalability and optimize network performance in various IoT deployment scenarios by replacing the ADR mechanism with these algorithms.

The rest of this paper is structured as follows: Section II discusses some recent works about SF selection. In Section III, we explore the optimization algorithms that were used. Section IV presents our methodology and the results of simulations conducted to evaluate the performance of these algorithms. The analysis of simulation results and comparison between optimization algorithms is discussed in the same section. Finally, section V concludes the paper by summarizing key findings and proposing avenues for future research.

II. Related Works

Enhancing spreading factors in LoRaWAN networks has numerous practical benefits, including improved network efficiency, reduced latency, extended end device battery life, and increased scalability. These benefits translate into real-world advantages such as cost savings through minimized need for extra infrastructure, enhanced service quality, more informed deployment planning, and enabling more complex and diverse IoT applications. Accordingly, these achievements allow organizations to gain a competitive edge through improved, more efficient, and more sustainable network management.

In the field of LoRaWAN network optimization, several significant contributions have been made. For example, in [10], authors focus on factors influencing LoRaWAN Quality of Service (QoS), particularly in scenarios involving multiple gateways. The impact of payload length on PDR is explored, highlighting how increasing gateway numbers may negatively affect network performance, leading to significant packet loss and increased energy consumption.

In [11], the authors addressed the challenge of improving network scalability by proposing an alternative method for assigning SF values to nodes. Their approach aimed to enhance data delivery probability while allowing a slight increase in device consumption, as validated through simulations.

Furthermore, in [4], researchers optimized transmission parameters in high-density Smart City environments to mitigate collisions and improve success rates. By dynamically distributing SFs using optimization techniques, they achieved lower delays and higher success rates, facilitating more efficient resource utilization.

Moreover, inefficiencies in standard LoRaWAN adaptive data rate control mechanisms are addressed in [12] by introducing a congestion classifier and a modified adaptive data rate control scheme. The results demonstrated that incorporating congestion estimation improved data transmission efficiency in long-range IoT services, reducing the impact of network congestion on QoS.

Similarly, adaptive solutions are investigated in [13] for optimizing LoRaWAN parameter settings to minimize channel utilization and maximize the number of delivered packets. Simulations validated the effectiveness of their proposed method, showing improvements when compared to heuristic-based approaches.

Following the same context, a theoretical analysis is provided in [14] regarding LoRa throughput in uplink scenarios, considering various types of interference and SF allocations. Their analysis highlighted the impact of imperfect SF orthogonality on throughput, enabling the design of specific SF allocation mechanisms for further improvements.

Authors in [15] proposed a tree-based SF clustering algorithm for improving the capacity of multi-hop LoRa networks. By offloading traffic into subnets with specific SF allocations, their approach enabled parallel transmissions and significantly enhanced network performance. Dealing with the multi-hop aspect, [16] proposes a solution to enhance the reliability of the LoRaWAN network by allowing direct communication between nodes towards gateways to address signal attenuation and data transmission failures caused by dense infrastructure and enhance the reliability of LoRaWAN networks.

Furthermore, LoRa systems are analyzed in [17] to improve the average system Packet Success Probability (PSP) under the unslotted ALOHA random access protocol. In that study, an optimization problem for SF allocation was formulated, and a sub-optimal scheme was proposed to achieve maximum connectivity.

Authors in [18] investigated energy efficiency for uplink LoRa networks by jointly optimizing user scheduling, SF assignment, and power allocation. Their proposed algorithms achieved near-optimal performance in terms of energy efficiency, demonstrating the potential for significant improvements in network sustainability.

All these discussed works emphasize the importance of addressing technical issues in LoRaWAN networks to fully leverage the potential of IoT applications in various deployment scenarios. A thorough review of the literature reveals a common trend: many studies either perform calculations without using a realistic simulation environment or do not prioritize PDR optimization. This underscores the need to analyze the impact of SF values, which in turn influence various parameters such as data rate, energy consumption, and range, on PDR optimization using a realistic simulator. To bridge this gap, this paper utilizes different optimization algorithms to determine the optimal SF values for each device, thereby maximizing PDR using the ns-3 simulator. To validate the quality of the results, a comparison with the ADR mechanism for optimizing network performance is conducted.

III. Proposed Approach

In this research, three well-known nature-inspired optimization algorithms, GA, SA, and DE, are investigated to evaluate their potential to improve LoRaWAN performance. Each algorithm is applied to the SF selection problem with the objective of maximizing the PDR. The fitness of a given solution is assessed based on the final PDR value obtained through simulations using the ns-3 simulator. While other metaheuristics, such as Tabu Search (TS), were also tested, only these three algorithms are considered in this study. This selection is made not only to avoid unnecessarily extending the paper but also because GA, SA, and DE demonstrated superior performance in terms of final results compared to other techniques. The following sections provide a brief description of these algorithms and their integration into LoRaWAN.

A. Optimization Algorithms

1) Genetic Algorithm (GA): GA [19] is a metaheuristic inspired by the principles of evolutionary biology. It employs a population-based search where potential solutions, represented as chromosomes, evolve over successive generations to improve solution quality. GA iterates through selection, crossover, and mutation operations to generate and refine new populations.

2) Simulated Annealing (SA): SA [20] is a probabilistic technique for optimization based on the annealing process in metallurgy. As applied to SF selection, it tries to maximize the Packet Delivery Ratio (PDR) by minimizing an energy function as its inverse. The algorithm searches for neighboring solutions and sometimes accepts poorer ones in order to move out of local optima, with this probability diminishing as the temperature is slowly reduced.

3) Differential Evolution (DE): DE [21] is a robust, population-based optimization method that has proven very effective in complex problem spaces. When choosing SF, it generates candidate solutions by using mutation, crossover, and a greedy selection operator that only retains the best individuals. Its differential mutation operation achieves a good balance between exploration and exploitation, pushing convergence towards optimal solutions.

B. Optimization Process

This study addresses SF optimization to improve PDR using optimization algorithms and network simulations, outlining the iterative process from configuration generation to evaluation and refinement.

The optimization process involves several key steps that are repeated iteratively until the optimal solution is found or the maximum number of evaluations is reached. We outline each step in detail as follows:

1) Step 1: Generating the Configuration Vector: The first step in solving the optimization problem is the generation of a configuration vector. This vector represents a specific set of SF values for the devices in the LoRaWAN network.

- Initialization: The optimization algorithm initializes a population of configuration vectors if it is the first iteration.
- Vector Representation: Each configuration vector is represented as a one-dimensional array of length n, where n denotes the number of end devices in the network. Each element in the vector corresponds to the SF value assigned to a specific device. Formally, the vector consists of decision variables $D = (D_1, D_2, \ldots, D_n)$, where each D_i represents the SF value assigned to end device i. In the context of LoRaWAN, SF is an integer ranging from 7 to 12. Fig. 1 illustrates the structure of the employed vector representation.

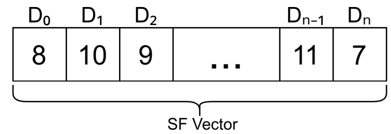


Fig. 1: Solution representation

- Population Update: The population of solution vectors is updated to explore new potential solutions. This could involve creating new vectors or modifying existing ones based on the performance of previous evaluations.

2) Step 2: Evaluation Function: A configuration vector is passed to the evaluation function to determine its effectiveness in terms of the target metric, which in our case, is the PDR.

- Sending SF Values: The generated SF configuration vector is sent to the evaluation function. This function interfaces with the ns-3 simulator to run a network simulation using the specified SF values.
- Running the Simulation: The ns-3 simulator executes the simulation using the given SF configuration. It models the network's behavior and computes the PDR based on packet transmissions and receptions within the simulated environment.
- Result Collection: ns-3 outputs the results, specifically the PDR for the given SF configuration. The evaluation function captures these results and processes them to extract the PDR.

3) Step 3: Returning Results to the Optimization Algorithm: The PDR value is returned to the optimization algorithm while ensuring the following tasks:

- Feedback Loop: The optimization algorithm receives the PDR value associated with the evaluated SF configuration and use them to assess the performance of the current configuration. It adjusts its search strategy to focus on more promising areas of the solution space.
- Refinement and Iteration: This phase involves generating new configuration vectors or modifying existing ones, guided by the evaluation results. The process repeats, with the algorithm continuously generating, evaluating, and refining SF configurations over multiple iterations.
- Convergence: The algorithmic cycle continues until it reaches the maximum number of evaluations or converges to an optimal solution. Convergence is typically defined as a point where further iterations yield negligible improvements in the PDR.

The diagram shown in Fig.2 represents the optimization process, illustrating the cycle of generating configuration vectors, evaluating them using the ns-3 simulator, and refining the optimization algorithm based on the evaluation results. Solving optimization problems in complex network scenarios, such as optimizing SF values in LoRaWAN networks, requires a structured and iterative approach. By generating configuration vectors, evaluating them through detailed simulations, and

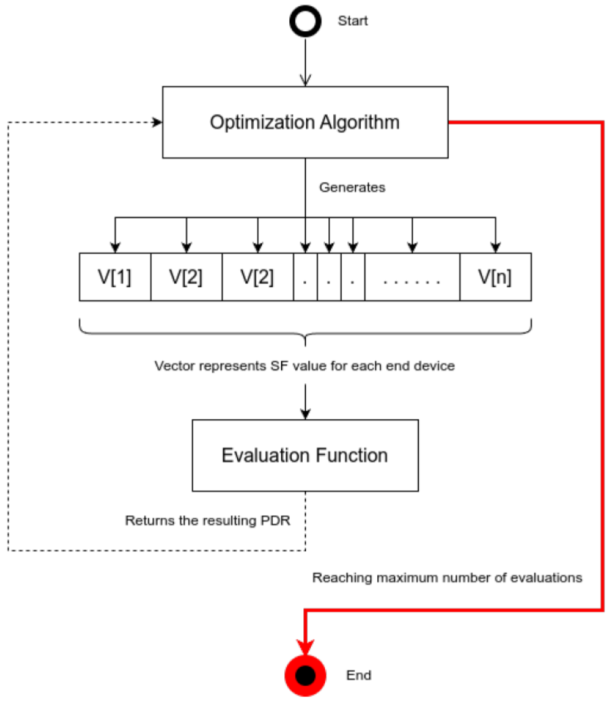


Fig. 2: Global Solving Optimization Process

TABLE I: Simulation Parameters

Parameters	Value
Algorithms	GA, SA, DE and ADR
Number of EDs	from 100 to 500 by step of 100
Payload length	from 10 to 50 bytes by step of 10
Radius	7.5 km
Simulation Time	60 min
App-period	packet every: 1, 5, 10, 15 minutes

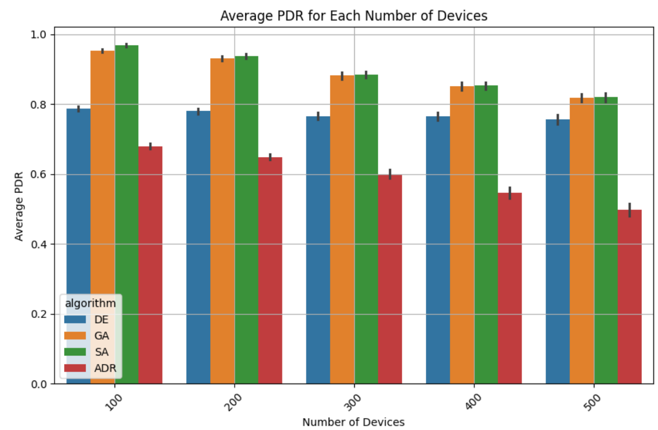


Fig. 3: Global Results

using the results to refine the search process, we can effectively navigate the search space to find optimal configurations. This method, facilitated by the integration of powerful optimization algorithms and robust simulation tools like ns-3, demonstrates the potential for achieving significant performance improvements in wireless communication networks. The detailed process ensures that we comprehensively explore possible configurations, ultimately leading to the identification of the best SF values that maximize the PDR in various LoRaWAN scenarios.

IV. Simulations and Results Analysis

A series of simulations across various scenarios were conducted to evaluate the performance of optimization algorithms in selecting SF values for LoRaWAN devices. The used simulator is ns-3 with the LoRaWAN implementation [22]. Each simulation case is defined by parameters such as the algorithm used, the number of devices in the network, payload size, and application period, which is the packet generation frequency.

A. Simulation Cases

We explored 400 simulation cases, comparing three metaheuristics and the ADR mechanism across different network sizes, payload sizes, and application periods. Table 1 summarizes all the simulation parameters used.

Ten simulation runs were conducted for each combination of algorithm, network size, payload size, and application period to ensure statistical robustness. The same set of scenarios was performed with ADR alongside a 24-hour simulation time to adjust SF values.

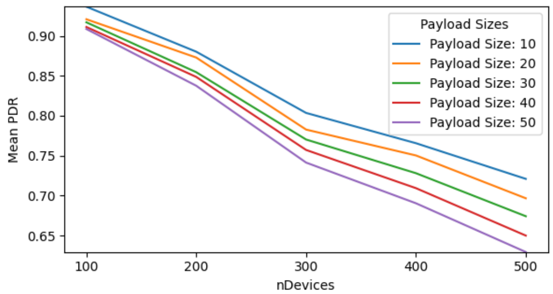
The computational experiments were carried out using resources of the High-Performance Computing Center of the University of Strasbourg. Furthermore, we used the MEALPY Python library [23], which includes GA, SA, and DE metaheuristic optimization algorithms, to optimize SF selection in LoRaWAN networks. The parameter settings for each metaheuristic experimentation have been set as the default values of the MEALPY library.

B. Discussing results

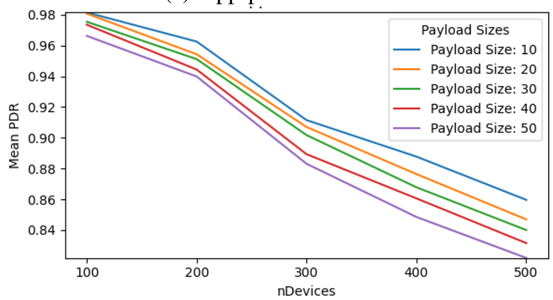
In this subsection, we discuss the overall results of the different cases, mainly regarding the PDR metric. Afterwards, we focus specifically on comparing ADR with SA while varying other parameters like payload size and application period.

1) Global Results:

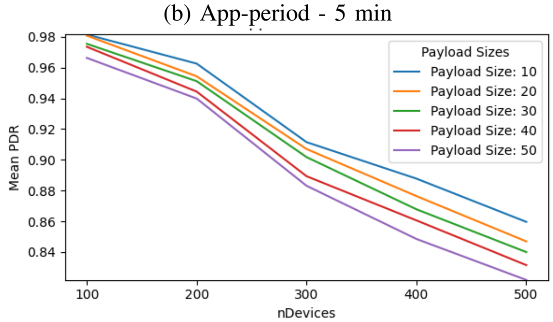
Fig. 3 shows the average PDR values for each algorithm for all packet generation frequencies and payload sizes while varying the number of devices. We can observe that both GA and SA algorithms perform well compared to the DE and ADR mechanisms, with SA having a slight edge. Its ability to adapt to varying network conditions and minimize packet loss was evident, especially in scenarios with high device density and packet-sending frequency. The advantage of SA over GA and DE can be explained by the presence of many local optimal solutions. Indeed, in such a scenario, SA's probabilistic acceptance of worse solutions allows it to escape local minima more effectively. These findings confirm the results of several studies such as in [19], [24].



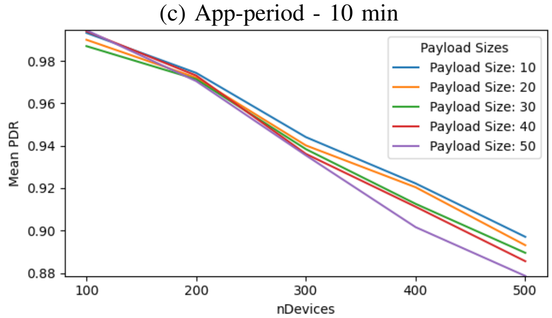
(a) App-period - 1 min



(b) App-period - 5 min



(c) App-period - 10 min



(d) App-period - 15 min

Fig. 4: Performance results of the SA algorithm while varying the payload size with different App-period values.

Accordingly, only SA is compared with the ADR mechanism in the upcoming experimentation. We can see also from Fig. 3 that as the number of devices increases, the performance of all algorithms declines. This is understandable since SF values assigned to end devices are treated as individual decision variables. Therefore, obtaining satisfactory final results in terms of PDR becomes challenging with a large number of decision variables. Moreover, increasing the number of nodes leads to network overload and a higher probability of collisions and interference.

*2) **Pairwise comparison between SA and ADR results**:* Following Fig. 4, it is clearly seen that SA exhibited a significant advantage in terms of final results. First, comparing the four sub-figures, it is noticed that a high number of devices results in more packet collisions, lowering PDR. Similarly, increasing the frequency of sending a packet results in low PDR scores. This latter can be clearly observed from Fig. 4a with an App-Period (packet generation frequency) of 1 min where the PDR values are 9% to 23% less than the one shown in Fig. 4d with an App-Period of 15 min. Furthermore, Fig. 4a, 4b, 4c and 4d show that bigger payload sizes affect PDR by 16% in some cases. All the figures also indicate that using a payload size of 10 results in better PDR.

Following Fig. 5, it is evident that the ADR algorithm's performance is significantly influenced by the number of devices, App-periods, and payload sizes. Increasing the number of devices consistently lowers the PDR due to more packet collisions, with the most substantial declines seen as device numbers approach 500. More frequent packet transmissions, as presented in Fig. 5a, result in lower PDR values, approximately 9% to 23% less than those observed with a 15-minute App-period as showed in Fig. 5d. Furthermore, larger payload sizes negatively impact PDR, with payload size 50 causing up to a 16% drop in some cases.

It can be seen that SA demonstrates a significant advantage over ADR, especially in handling higher device counts and frequent transmissions. Although both algorithms performance deteriorate when increasing devices and larger payload sizes, SA maintains higher PDR values overall. ADR shows a significant drop in PDR, particularly with a 1-minute App-period and larger payload sizes, indicating higher sensitivity to these factors compared to SA.

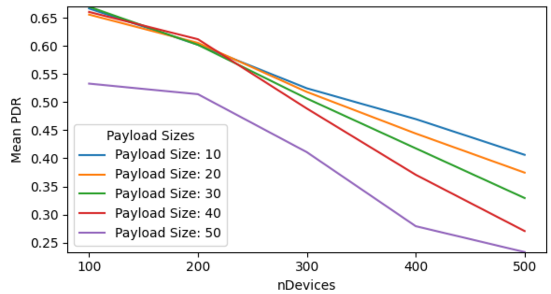
V. Conclusion

An optimization-based approach is presented to optimize PDR in LoRaWAN networks by dynamically selecting SF values using GA, SA, and DE. They are used as alternatives to the ADR mechanism, where the objective was to improve SF allocation, improve network scalability, and optimize overall performance in different IoT deployment scenarios. Through extensive simulations and comparative analysis, the optimization algorithms demonstrated significant improvements in PDR compared to the conventional ADR mechanism. In particular, SA exhibited superior stability and consistency in performance, especially in dynamic environments characterized by fluctuating interference levels.

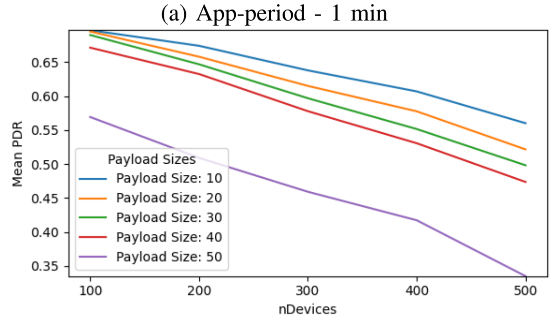
Future works could explore adaptive optimization methods responsive to real-time conditions, consider metrics like capacity and collision probability, and extend the approach to large-scale, multi-gateway LoRaWAN networks for better scalability.

VI. Acknowledgment

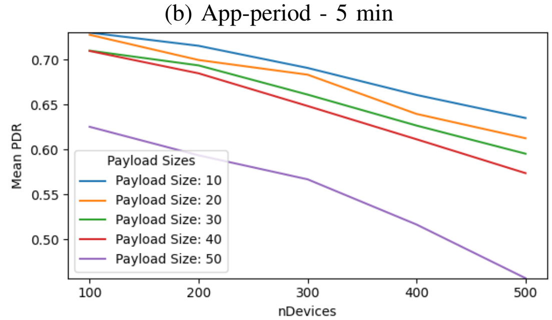
All computational experiments were conducted at the High-Performance Computing Center of the University of Stras-



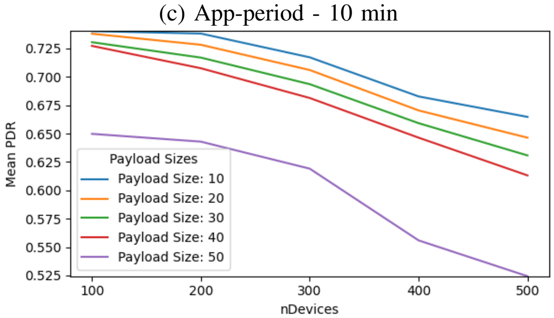
(a) App-period - 1 min



(b) App-period - 5 min



(c) App-period - 10 min



(d) App-period - 15 min

Fig. 5: Performance results of the ADR mechanism while varying the payload size with different App-period values.

bourg, funded by Equip@Meso and CPER Alsacalcul/Big Data.

References

[1] M. A. M. Almuhaya, W. A. Jabbar, N. Sulaiman, and S. Abdulmalek, "A survey on lorawan technology: Recent trends, opportunities, simulation tools and future directions," *Electronics*, vol. 11, no. 1, 2022.

[2] G. H. Derévianckine, A. Guitton, O. Iova, B. Ning, and F. Valois, "Opportunities and challenges of lora 2.4 ghz," *IEEE Communications Magazine*, 2023.

[3] C. Caillouet, M. Heusse, and F. Rousseau, "Bringing fairness in lorawan through sf allocation optimization," in *2020 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1–6, IEEE, 2020.

[4] A. L. Emmanuel, X. Fernando, F. Hussain, and W. Farjow, "Optimization of spreading factor distribution in high density lora networks," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pp. 1–5, May 2020.

[5] S. Mnguni, P. Mudali, A. M. Abu-Mahfouz, and M. Adigun, "Performance evaluation of spreading factors in lora networks," in *Towards new e-Infrastructure and e-Services for Developing Countries* (R. Zitouni, A. Phokeer, J. Chavula, A. Elmokashfi, A. Gueye, and N. Benamar, eds.), (Cham), pp. 203–215, Springer International Publishing, 2021.

[6] A. Semtech, "Understanding the lora adaptive data rate," *Available on: https://loradevelopers. semtech. com/documentation/tech-papers-and-guides/understanding-adr/.(Accessed on 23.03. 2022)*, 2019.

[7] A. Hoeller, R. D. Souza, S. Montejo-Sánchez, and H. Alves, "Performance analysis of single-cell adaptive data rate-enabled lorawan," *IEEE Wireless Communications Letters*, vol. 9, no. 6, pp. 911–914, 2020.

[8] J. Moraes, H. Oliveira, E. Cerqueira, C. Both, S. Zeadally, and D. Rosário, "Evaluation of an adaptive resource allocation for lorawan," *Journal of Signal Processing Systems*, vol. 94, pp. 65–79, Jan 2022.

[9] K. S. Abakar, I. Bennis, A. Abouaissa, and P. Lorenz, "A hidden parameter study for traffic-oriented lorawan deployment," in *2024 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 1376–1381, 2024.

[10] K. S. Abakar, I. Bennis, A. Abouaissa, and P. Lorenz, "A multi-gateway behaviour study for traffic-oriented lorawan deployment," *Future Internet*, vol. 14, no. 11, 2022.

[11] A. Tiurlikova, N. Stepanov, and K. Mikhaylov, "Method of assigning spreading factor to improve the scalability of the lorawan wide area network," in *2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pp. 1–4, Nov 2018.

[12] D.-Y. Kim, S. Kim, H. Hassan, and J. H. Park, "Adaptive data rate control in low power wide area networks for long range iot services," *Journal of Computational Science*, vol. 22, pp. 171–178, 2017.

[13] J. Moraes, N. Matni, A. Riker, H. Oliveira, E. Cerqueira, C. Both, and D. Rosário, "An efficient heuristic lorawan adaptive resource allocation for iot applications," in *2020 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1–6, July 2020.

[14] A. Waret, M. Kaneko, A. Guitton, and N. El Rachkidy, "Lora throughput analysis with imperfect spreading factor orthogonality," *IEEE Wireless Communications Letters*, vol. 8, pp. 408–411, April 2019.

[15] G. Zhu, C.-H. Liao, T. Sakdejayont, I.-W. Lai, Y. Narusue, and H. Morikawa, "Improving the capacity of a mesh lora network by spreading-factor-based network clustering," *IEEE Access*, vol. 7, pp. 21584–21596, 2019.

[16] M. A. H. Amor, I. Bennis, K. Saleh Abakar, A. Abouaissa, and P. Lorenz, "Llnrm: Lorawan loss node relay mechanism for smart cities," in *2025 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2025.

[17] J.-T. Lim and Y. Han, "Spreading factor allocation for massive connectivity in lora systems," *IEEE Communications Letters*, vol. 22, pp. 800–803, April 2018.

[18] C. Caillouet, M. Heusse, and F. Rousseau, "Optimal sf allocation in lorawan considering physical capture and imperfect orthogonality," in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec 2019.

[19] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–73, 1992.

[20] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[21] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, Dec 1997.

[22] D. Magrin, M. Capuzzo, and A. Zanella, "A thorough study of lorawan performance under different parameter settings," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 116–127, 2019.

[23] N. Van Thieu and S. Mirjalili, "Mealpy: An open-source library for latest meta-heuristic algorithms in python," *Journal of Systems Architecture*, 2023.

[24] Z. Michalewicz, *How to Solve it: Modern Heuristics*. Springer Science & Business Media, 2013.