Research Article

Nabaa G. Adiel*, Asia A. Salman, and Nidaa F. Hassan

Node failure in self-organized sensor networks

https://doi.org/10.1515/eng-2024-0083
received April 27, 2024; accepted August 03, 2024

Abstract: Wireless sensor networks (WSNs) encountered substantial obstacles in contexts characterized by frequent sensor node failures. Overcoming these obstacles requires a remedy that not only identifies node failures but also improves network self-organization. This work introduces a method that merges the Cuckoo Search Optimization algorithm (CSO) with the suggested Guided and Effective Search (GES) algorithm to improve the network's ability to self-organize and maintain efficiency during node failures. The method combines CSO's search capability for finding node configurations with GES' effectiveness in local searches within the network structure. Together, they establish a system for fault detection network optimization, and improve self-organization, ensuring that the network could adapt and withstand disruptions. Comprehensive simulation results demonstrated the method's superiority compared to the existing methods. The system demonstrates enhancements in fault detection accuracy, network self-organization, packet delivery rate, and overall energy efficiency. In addition, the simulation results highlight the improved performance of the combined approach compared to the Particle Swarm Optimization algorithm. Integrating CSO and GES marked advancement in creating self-organizing WSNs offers reliability and longevity for networks used in critical applications.

Keywords: wireless sensor networks, self-organization, node failures, time to live, sink node, average task success rate

1 Introduction

Wireless sensor networks (WSNs) are critical for monitoring various environmental factors and are widely used in applications such as agriculture, weather tracking, and industrial operations [1]. These networks operate autonomously, organizing themselves and using strategically placed sensor nodes to gather and transmit data [2]. WSNs control and monitor aspects such as sound, temperature changes, pollution, waves, and wind. They are useful in real-time, like in agriculture monitoring, weather tracking, and surveillance of solar plants and factories [3,4].

A wireless sensor is defined as a device with computational and power provisions for carrying out the process of interfacing between users and the physical world through a computer [5]. The core components of a sensor node include a radio transceiver with an antenna for communication purposes, a microcontroller for processing data received from sensors, an interface circuit for integrating sensor data streams, and typically a battery as the power source [6,7]. This setup allows the wireless sensor to capture data efficiently and bridge the gap between the physical and digital realms [8].

WSNs have substantial problems, especially in contexts characterized by frequent sensor node failures, notwithstanding their usefulness [9]. These failures may arise due to several conditions, such as battery depletion, hardware faults, and environmental disturbances [10]. Malfunctioning sensor nodes result in the direct transmission of data to the sink, resulting in energy wastage. Node errors in WSNs may be categorized into two main groups: software faults, which occur when the system software of a node is incorrectly designed, and hardware faults, which occur when various hardware components of a node are broken [11]. If the WSNs can identify and manage defective nodes and data accurately, then WSNs can offer dependable performance and high-quality data to the end users. A high degree of accuracy is required in the fields like environment, agriculture, and health [12].

The first issue in WSNs is the extendibility of the system functionality and efficiency, given high node mortality rates. Certain presented techniques do not possess aspects such as fault tolerance, adaptability, and the ability to cope with all dynamism in the network settings. Most of the architectures involve control dependencies, which are considerably restrictive in boosting scalability and compromising the networks' stability. Besides, many of these techniques may require relatively large computational activity and exchange of messages between nodes, which can be prohibitive in WSNs.

*** Corresponding author: Nabaa G. Adiel,** Department of Computer Science, University of Technology, Baghdad, Iraq, e-mail: cs.22.24@grad.uotechnology.edu.iq
Asia A. Salman: Department of Computer Science, University of Technology, Baghdad, Iraq, e-mail: 10035@uotechnology.edu.iq
Nidaa F. Hassan: Department of Computer Science, University of Technology, Baghdad, Iraq, e-mail: 110020@uotechnology.edu.iq

The primary challenge addressed in this study is the efficient distribution of tasks and optimization of paths within WSNs, especially when faced with node failures. Conventional methods often struggle to maintain performance and resilience under such conditions. This research aims to overcome these limitations by introducing a self-organizing framework that seamlessly integrates Guided and Effective Search (GES) with Cuckoo Search Optimization (CSO). This innovative approach ensures robust task distribution and path optimization, maintaining high task success rates and network resilience even when a significant number of nodes are disabled, thus simulating real-world scenarios of hardware failures and environmental disturbances.

This study underscores the importance of self-organization in creating network structures that can enhance efficiency independently and effectively tackle challenges.

The key contributions of this research include:
- Introducing a combined framework that integrates CSO and GES for enhanced network performance.
- Evaluating the suggested system frameworks in improving fault detection accuracy, network self-organization, and task delivery rates.
- Demonstrating the framework's ability to maintain network efficiency and robustness in the presence of node failures through comprehensive simulations.

The structure of this article is organized as follows: Section 2 describes related works. Section 3 focuses on self-organization mechanisms within WSNs. Section 4 presents the suggested system framework, detailing the components and processes involved. Section 5 explains the proposed framework and includes pseudo-code to illustrate its implementation. Section 6 describes the experimental setup and performance metrics. Section 7 presents the simulation results and the discussion. Finally, Section 8 concludes the article.

2 Related works

In WSNs, several limitations have served as the basis for the creation of improved approaches. Several current approaches exhibit deficiencies in fault tolerance, presenting a substantial obstacle in guaranteeing network operation in the face of node failures in real-world applications. These flaws result in bottlenecks with centralized control centers, decreasing scalability and reliability. However, the existing works fail to address the requirements of networks' different states and abnormal node issues in complex and dynamic conditions and, therefore, have some limitations in their efficiency. Another limitation that applies to many suggested solutions is that high computation requirements, together with substantial communication costs, are not easily achievable in resource-scarce WSNs. All these constraints advocate for the need to develop highly scalable, robust, and flexible solutions in WSNs.

Gutiérrez and Ponce presented artificial hydrocarbon networks in 2019 and applied them, and learned about the failures of WSN sensor nodes in damp indoor studio environments [13]. This approach is to usesupervised learning in which it learns the necessary output from the real sensor data and the specifications of the web service before it asks questions concerning temperature and faults in the sensor. The study reported 94% test data recovery. The next improvement is to fully comprehend the enhancements in the dynamic operation algorithm and test it on larger and more advanced WSNs.

Bista and Choudhary [14] proposed a novel method for fault detection, utilizing Spearman's rank correlation coefficient and K nearest-neighbor algorithm for classification. Consequently, the ANCDFD model outcompetes the metric correlation-based distributed fault detection (MCDFD) with respect to the accuracy of detection and false-positive rates, proving its essentiality in the analysis of the node status field.

The contribution presented by Palanikumar and Ramasamy in 2019 [15] was a method of matrix calculus, which they invented for the identification of nodes in WSNs. The strategy that they employed included finding the rows and columns in the faulty nodes in RTPs, which allows for detecting the multiple faulty node problems and results in delay comparisons. This optimized analyzing the network health compared with the time measurements of round trip delay.

Zidi and colleagues [16] demonstrated the application of support vector machines (SVMs) in recognition of flaws in WSNs. This seems to be a viable method as SVM was able to categorize the sensor activities with minimal use of resources. The technique stands up through the application of the statistical learning theory, which states a decision-making process, the main goal of which is to show the efficiency of the diagnosis in those areas where accuracy and speed are required. This approach stands out by meeting cluster leaders' requirements not only with the ease of filtration but also by ensuring exacting detection rates are concurrently assured.

Jia *et al.* [17] claimed that the LEFD mechanism offers solutions to mitigate the problems of finite energy sources while improving the faults of the WSNs. From this new approach, time and spatial positioning data are being used to sensibly determine which fault to detect without the need to correlate neighboring button pushes, and the whole network's energy is managed. Their method, as opposed to the other methods, tackles problems like hardware faults, energy distribution among nodes, and security breaches. The method thus provides a solution that is engaging and enhances transmission as well as lowers energy consumption in WSNs.

Satyanarayana *et al.* [18] devised an algorithm that aims at detecting quiescence in sensor networks (WSNs) while minimizing cost and maximizing network coverage. They resolved this in a unique way: they used the relay nodes as the points for the positioned sensor nodes, which was different from others; the strategy was a two-stage process, the intra- and inter-segmentations. The objective of the approach was the extension of network throughput by making use of nodes and physical proximity facilitating better analytics and network administration than the conventional considered ways.

Wu and colleagues [19] proposed a smart method, called self-organizing map (SOM) trend correlation detection (TCD) that can be utilized for detecting faults in WSNs. The technique employed both the SOM and the TCD to cluster nodes that have the same data correlation, taking advantage of TCD to find faults quickly in the recognized groups of individuals. This scheme was capable of achieving accuracy greater than 95% with respect to outlier and random defects in WSN. Unlike existing methods, SOM TCD provides a fault detection solution that becomes consistent even when faced with changed rates.

Umamaheswari and Antony [20] put forward a method to detect and rectify failures in WSNs by incorporating binary and non-binary feedback mechanisms. This approach surpassed monitoring by reducing communication overhead by 80% and achieved a 95% accuracy rate. It utilized the AODV routing protocol, bolstered with binary and non-binary strategies for fault identification and correction, leading to improved delivery rates, reduced routing overhead, and minimal end-to-end delays. The integration of an AES DES encryption algorithm further enhanced network security against access attempts, ensuring the secure transmission of data. This strategy offered a solution for enhancing the resilience of WSNs in the face of node failures through a blend of fault detection methods and security protocols.

3 Wireless networks and self-organization technique

Self-organization plays a role in networks, particularly in sensor networks, as it is essential for achieving high levels of reliability and efficiency in dynamic environments [21]. A self-organization system can adapt dynamically whenever the surrounding conditions demand it without any interference. The origins of self-organization frameworks can be traced back to physics social groups and the study of insects [22]. The concept highlights the importance of fault tolerance and network robustness by leveraging self-organizing features. These advancements enable decision-making [23], emphasizing the ability of WSNs to adapt and maintain connectivity even in the face of node failures. By integrating self-organization, WSNs exhibit a capacity to adjust dynamically to changes and disruptions ensuring service and data integrity crucial for various applications relying on these networks [24].

4 Suggested system framework

The system framework outlines a structured approach to construct and maintain a robust and efficient WSN. This framework leverages the principles of directed graphs for network construction and incorporates self-organization mechanisms to dynamically adapt to node failures and optimize network performance. The following sections describe the key components of the framework in detail.

4.1 Created sensor network

The construction of a WSN starts with a central node (the sink node), which serves as the core of the network. Then, more nodes are gradually added one by one until the limited number of nodes or the network size (M) is reached. However, the maximum network size (M) is specified in this work only for simulation purposes. The parameter M can be changed to any value. Also, the created network specifies other parameters, such as the number of connections for each node, time to live (TTL) for each task, and triggering conditions for self-organization, as explained below.

Hence, the directed graphs approach, as shown in Figure 1, is used until there are a total of M nodes in the network.

4.2 Self-organization in the proposed system framework

In this work, self-organization is the target aspect, as it enables the dynamically created network to adapt to node

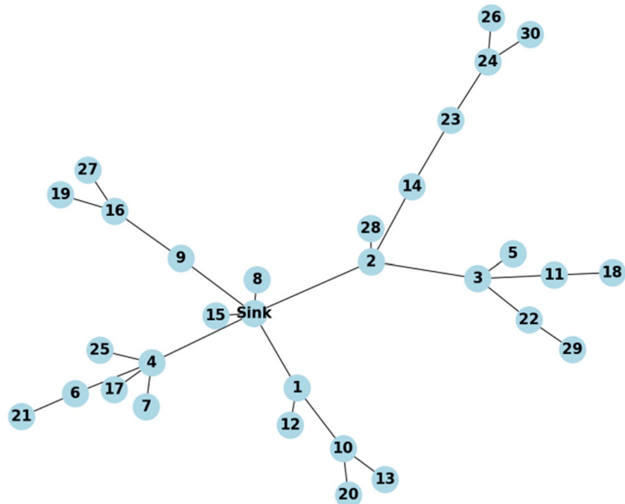

Figure 1: Graphical representation of WSN topology.

failures and optimize its performance. In this work, triggering conditions are set to heuristically decide which algorithm can be adapted to find the path that can be used for task distribution, as shown in (algorithm0). The heuristical behavior is to monitor the dynamic environment (WSN). The process involves monitoring node failures, triggering optimization algorithms, and selecting the most efficient paths for task distribution. Each node $\boldsymbol{i}$ maintains a failure counter $\boldsymbol{C_i}$ to monitor the status of its connected neighbors.

In the proposed framework, the self-organization process includes the following (algorithm 0):

Algorithm 0: Embedded self-organization process

Monitor node failures:

Calculate the failure count for each node $\boldsymbol{i}$.

$\boldsymbol{C_i} = \sum_{j \in \text{neighborhood}(i)} \boldsymbol{\delta_j}$

where $\boldsymbol{\delta_j} = 1$ if node $\boldsymbol{j}$ failed, and $\boldsymbol{\delta_j} = 0$ otherwise.

Triggering self-organization:

If the failure count $\boldsymbol{C_i}$ for a node $\boldsymbol{i}$ is at least half of its neighborhood:

$C_i \geq \frac{\text{Number of neighborhoods(i)}}{2}$

Else continue using GES.

5 Simulation model scenario

In this section, a detailed description is shown of the combined algorithms CSO with GES. After creating the network, the node failure problem is simulated in the created framework. Triggering self-organization leads to the use of the combined approach, which aims to improve the network's throughput, fault tolerance, and task transmission efficiency. A detailed explanation of the proposed scenario, along with the corresponding pseudo-code, is shown:

Pseudo-code of proposed scenario

1. Start.
2. Construction of the network.
3. Set initial parameters by defining total_tasks, success_count, TTL values, number of neighborhoods, and failure ratio.
4. Task distribution using GES (go to algorithm 1):

For each task:

- Select source and destination nodes.
- If the destination is active and a path exists:
- Calculate the path length.
- Decrement TTL for each hop.
- If TTL ≥ 0, increment success_count.

5. Monitor node failures and trigger self-organization:

Monitor node failures:

- Calculate the failure count for each node i.

Trigger self-organization:

- If the failure counts C_i for a node, i is at least half of its neighborhood.
- Evaluate paths using GES: Check paths to avoid failed nodes.

Optimize paths using CSO (go to algorithm 2): optimize paths to find the most efficient route free of failed nodes.

- Else, continue using GES.

6. Evaluate network performance.
7. End

5.1 GES

The GES algorithm includes features to enhance task distribution efficiency under dynamic network conditions (see Algorithm 1).

Algorithm 1: Task distribution using GES

Start

Set 'total_tasks' to 100. Initialize 'success_count' to 0.

For each task in 'total_tasks'

    Select source and destination:

        Set the source to "Sink".

        Choose a destination node randomly from nodes in graph G.

        Check if the status of the selected destination node is active or not.

Algorithm 1: Task distribution using GES

Check path existence, length, and TTL:
If there is a path from source to destination in G:
Calculate the shortest path length between the source and destination.
For each hop along the path to the Destination decrement TTL by 1 (NEW_TTL = TTL -1)
Evaluate task success within TTL limits:
Check IF (NEW_TTL ≥ 0), then increment success_count by 1.
End

5.2 Cuckoo swarm optimization (CSO)

The CSO algorithm was used to find optimal solutions to complex problems by mimicking the brood parasitism behavior of cuckoos. In WSNs, CSO optimizes network paths and enhances task transmission efficiency by iteratively evaluating and improving path fitness using the following fitness function:

$$\text{Fitness}\ (P_i) = \frac{1}{L_i + K \cdot F_i}, \quad (1)$$

where L_i is the length of path i (number of hops), F_i is the number of failed nodes along path i, and k is a penalty factor that increases the impact of failed nodes on the fitness score, especially in the presence of node failures. This algorithm enhances the self-organization and fault tolerance of WSNs, ensuring robust and efficient operation in dynamic environments.

Algorithm 2: Optimizing network paths with CSO

Start
Initialize paths with 'initial_paths'.
Identify the path with the best fitness, marking this as 'best_path' and recording its fitness as 'best_fitness'.
For each path in paths:
If the current path is not valid:
Generate a new path new_path from the current path
If new_path is valid:
Calculate the fitness of new_path.
If new_path's fitness is better than the current path's fitness:
Update the current path in 'paths' with 'new_path'.
Update the fitness score for this path
If new_path's fitness is also better than 'best_fitness':
Update best_path with new_path.
Update best_fitness with new_path's fitness.
end

6 Experiments

The experiments were crafted to evaluate the efficacy of the self-organization technique introduced in this work. This framework technique combines GES with CSO after heuristically deciding which path to follow in order to enhance task distribution.

6.1 Experimental setup

The WSN was represented using a directed graph termed G, consisting of (M) nodes, and the communication links between them were set as a setting parameter (N). To initiate tasks, a dedicated node called the "Sink" served as the hub for data collection, which is the first node in the network, and the network gradually expanded its size to 100 nodes and then to 200 nodes. Both of these two WSN sizes were studied and analyzed, adhering to a connectivity rule that limited each node to a maximum of N connections. This network setup was designed to simulate real-world WSN structures while maintaining a level of complexity.

6.2 Task dispatching process

Tasks were dispatched from the "Sink" hub node to other nodes within the network through a series of simulations that managed the transmission of tasks. The effectiveness of these transmissions depended on factors such as the paths taken, the operational status of the destination nodes, and compliance with TTL restrictions.

6.3 Modeling node failure problem

Simulating a node failure in the proposed WSN is a real-world challenge like hardware failures or environmental disturbances. In this scenario, 25% of the network nodes were intentionally disabled at random to show the problem of having unconnected paths (node falling). This approach triggered the self-organization technique to decide which algorithm is to be used to solve task distribution and path optimization. Evaluation of the network's resilience and effectiveness under such a problem is explained below.

Table 1: Simulation parameters

Parameter	Value
No. of nodes	100, 200
TTL	2, 5, 8
Task rate	100
No. of connections	4, 5
Failure rate	25%, 40%

6.4 Performance metrics

The performance of the proposed framework was assessed using key metrics that reflect the network's efficiency and resilience, particularly in the presence of node failures. The primary metrics used to measure performance were the average task success rate (ATSR) and average delay. The ATSR measures the ratio of successfully delivered tasks to the total number of tasks dispatched from the Sink node, as shown in Equation (2).

$$\text{ATSR} = \frac{\text{Success\_count}}{\text{Total\_tasks}}. \quad (2)$$

The average delay is defined as the average number of hops taken for a task to reach its destination from the source. This metric is significant, as it reflects the efficiency of the routing protocol in terms of the time taken to deliver tasks. It is given by Equation (3):

$$\text{Average delay} = \frac{\sum_{i=1}^{N}\text{Hops}_i}{N}, \quad (3)$$

where Hops_i represents the number of hops for the i-th successfully delivered task, and N is the total number of successfully delivered tasks.

7 Simulation results and discussion

The simulation, as outlined in Table 1, concentrated on evaluating the effectiveness of the combined algorithm known as GES alongside CSO under various network setups. It delved into the impact of changing the number of neighbors per node from four to five on the network's performance in task completion and its resilience to node failures.

Figure 2 illustrates the average success rates for GES, CSO, and combined strategies across three TTL values (2, 5, and 8) with a 25% node failure rate and a maximum number of 4 neighbors per node. Altogether, the combined strategy excelled in the performance of both GES and CSO concerning robustness and distribution of tasks. As the TTL value increased, the success rates improved for all strategies, with the combined strategy maintaining the highest performance. This indicates the combined strategy's effectiveness in leveraging both GES and CSO strengths, ensuring optimal path selection and fault tolerance.

To expand the simulation, the number of neighborhoods was increased to 5, as shown in Figure 3.

Increasing the number of neighbors from 4 to 5 led to raising the average success rates for all values of TTL and all strategies. Thus, all the GES, CSO, and combined strategies experienced an increased number of successful tasks due to the improved network connectivity.

For TTL = 2, the success rates increased substantially with the additional neighbors. The combined strategy, in particular, showed remarkable improvement, indicating its robustness in handling low TTL values even under node failures.

For TTL = 5, the trend of improvement continued. The combined approach nearly reached a 100% success rate

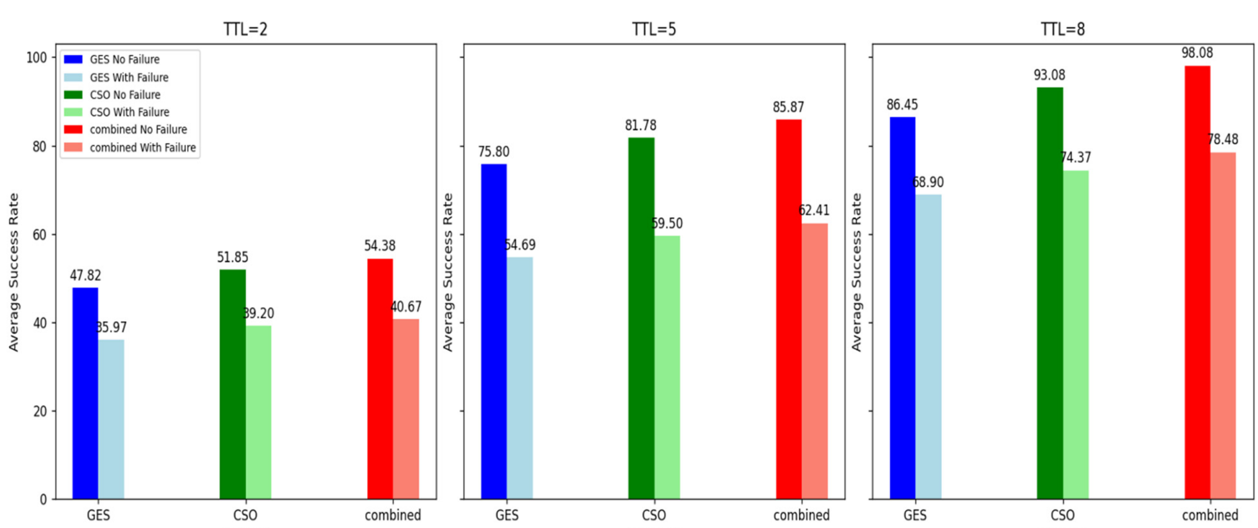

Figure 2: Simulation results with TTL = 2, 5, and 8, neighbors = 4, and 25% failure.

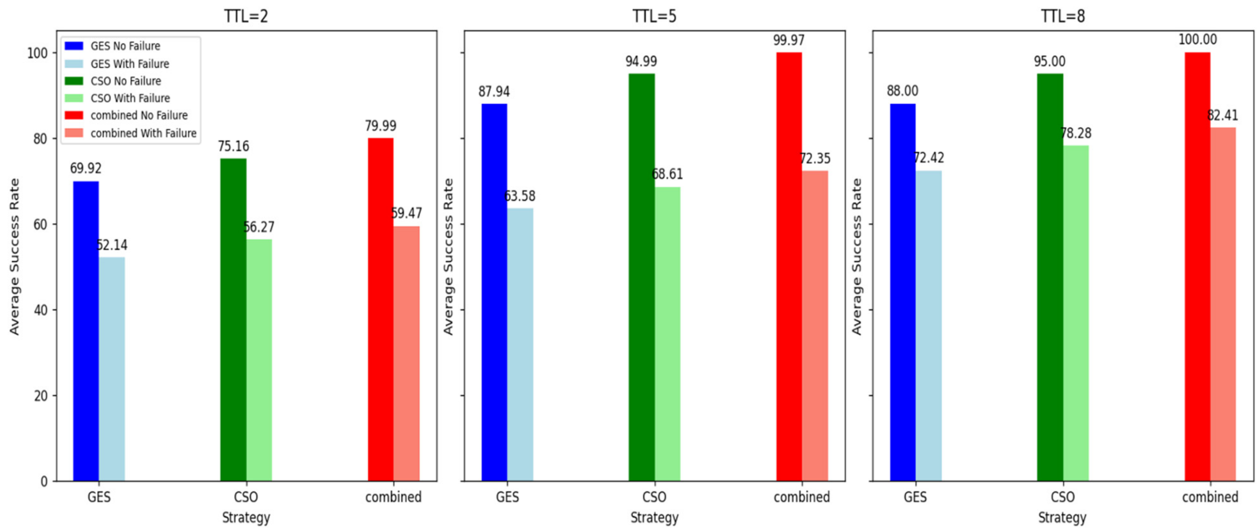

Figure 3: Simulation results with TTL = 2, 5, and 8, neighbors = 5, and 25% failure.

with no failures, demonstrating its superior ability to maintain network performance with increased connectivity. The GES and CSO strategies also showed significant improvements, with better handling of node failures.

For TTL = 8, the results further confirmed the benefits of higher connectivity. All strategies performed better with 5 neighbors compared to 4, with the combined strategy consistently achieving the highest success rates. The improvement in the success rate under failure conditions for the combined approach was particularly notable, indicating its enhanced fault tolerance.

The combined strategy leveraged both GES for task distribution and CSO for path optimization, resulting in shorter and more reliable paths even in the presence of node failures.

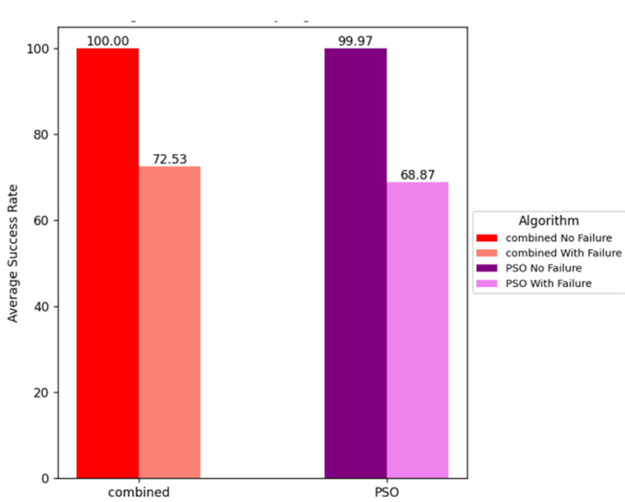

Figure 4: Comparison of average success rates for hybrid method and PSO.

To further validate the effectiveness of the proposed combined strategy, more simulations were run and compared to Particle Swarm Optimization (PSO).

The results presented in Figure 4 show the significantly improved performance of the hybrid approach under the specific conditions with a TTL value of 5, 100 nodes, a maximum of 5 neighborhoods, and a 25% node failure rate.

Under these conditions, the combined strategy demonstrates higher performance, consistently outperforming PSO. This notable improvement is attributed to the combined strategy's superior capability in optimizing transmission paths. The combined strategy creates more reliable and efficient paths than PSO by using the global search power of CSO and the precise local search power of GES. This leads to a lot more tasks being completed successfully and better network performance.

To enhance the accuracy of the specified combined strategy, the simulation was expanded to include 200 nodes, and the results were compared to those of 100 nodes with a failure rate of 40%. The results, as depicted in Figure 5, show the performance of the combined strategy under this condition.

Based on the simulation results for network sizes of 100 and 200 nodes, a detailed comparison and discussion can be derived as follows:

Path improvement: The combined algorithm has demonstrated a notable ability to enhance pathfinding and maintain high task success rates, even under increased node failures and larger network sizes.

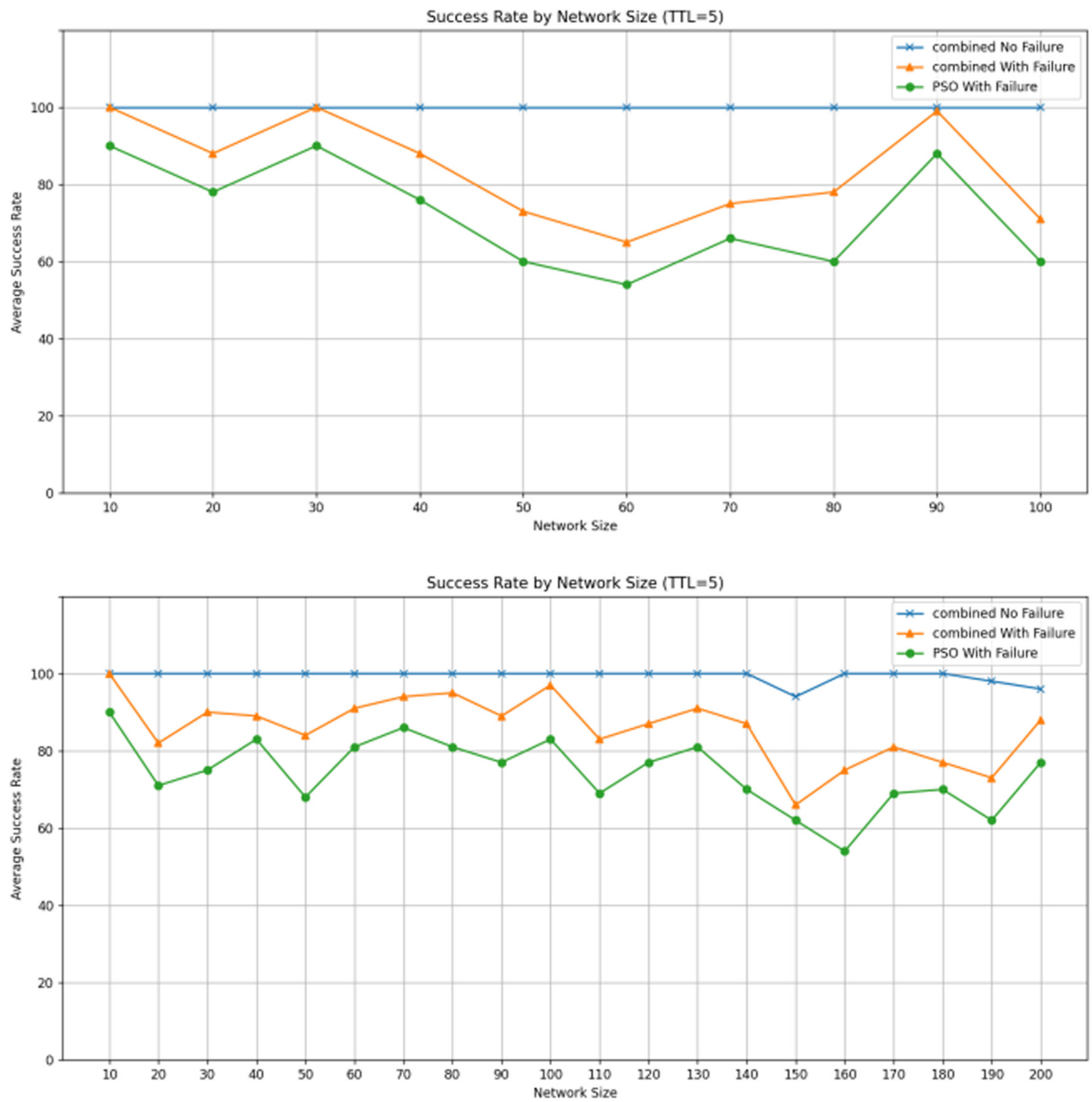

Figure 5: Comparative success rate analysis of combined algorithm and PSO with varying network sizes.

Failure resilience: Despite a significant node failure rate of 40%, the combined algorithm effectively maintained an average success rate close to 80–100% across both 100 and 200 node network scenarios. This indicates a robust path optimization mechanism that can dynamically adapt to network disruptions and continue to find viable paths for task delivery.

Scalability: In networks scaling from 100 to 200 nodes, the combined algorithm showed minimal degradation in success rates. It maintained near-perfect task delivery in no-failure scenarios and high success rates in failure scenarios, demonstrating its capability to scale effectively with network size.

The algorithm consistently surpassed the PSO algorithm in terms of task delivery in both scenarios with 100 and 200 nodes. This superior performance indicates that the combined algorithm is adept at handling the complexity and increased path lengths inherent in larger networks, ensuring that the maximum number of tasks are delivered successfully.

The average delay was also measured, and the results are shown in Figure 6.

As shown in Figure 6, in the case of the 100 nodes in the network, the combined algorithm with failures kept lower average delays than the PSO algorithm. This trend was further pursued with the network size increased to 200 nodes, and the net of the whole combined algorithm revealed the ability to handle the larger networks and failure rates.

The delay analysis also embodies this feature where the combined algorithm was proven to exhibit lesser delay

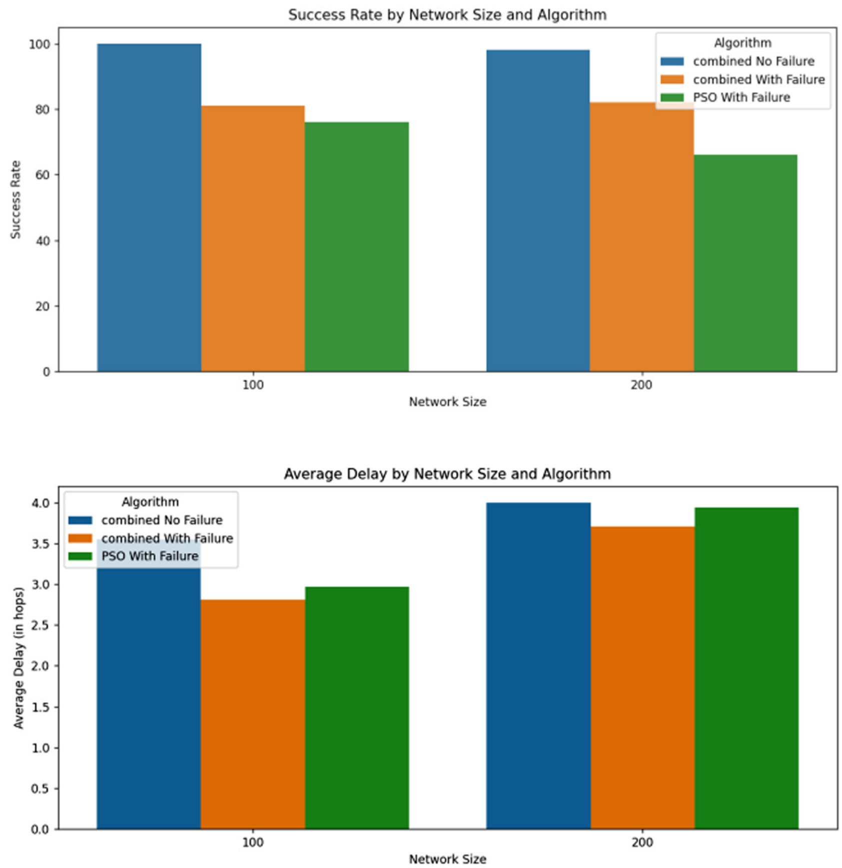

Figure 6: Comparative success rate and delay analysis of combined algorithm and PSO with varying network sizes.

across different networks' sizes than the other comparable algorithms, displaying its efficiency in task distribution and delay reduction. These results declare the combined algorithm's superior performance in enhancing network resilience and efficiency.

8 Conclusion

Due to the highly challenging environment, there is the necessity of implementing two fundamental measures: resilience and efficiency for WSNs. This research presents a new integrated architecture that combines GES with CSO to address the limitations of conventional task transmission and fault tolerance. The conclusions fully correspond to the potentialities of a comprehensively and self-organized high-level system capable of surmounting the problems inherent to real-life application.

In the middle of the tumult caused by node failures, the GES–CSO algorithm stood out as a very dependable solution. It skillfully managed the intricacies of networks with 100 and 200 nodes, continuously achieving excellent success rates despite a 40% node failure rate. This persistence is seen in its consistently high success rate, sustaining almost flawless performance. The GES–CSO combined approach surpassed the PSO method and established a higher benchmark for network efficiency.

Delving deeper, the analysis of average delay unveiled the algorithm's prowess in minimizing latency. The combined strategy ensured swift task delivery, keeping delays to a minimum even as the network scaled in size and complexity. Such efficiency in reducing delays underscores the framework's adeptness at optimizing network paths and maintaining seamless operation.

This article does not just present an algorithm but offers a vision of future-proof WSNs that are resilient, scalable, and efficient.

Funding information: The authors state no funding involved.

Author contributions: Nabaa Ghadeer wrote the manuscript with support from Asia Ali and Nidaa Flaih. Nabaa Ghadeer developed the theoretical formalism and performed the analytic calculations and simulations. Nabaa

Ghadeer carried out the experiment. Nabaa Ghadeer, Asia Ali, and Nidaa Flaih contributed to the final version of the manuscript. All authors have accepted responsibility for the entire content of this manuscript and given consent to its submission to the journal, reviewed all the results, and approved the final version of the manuscript.

Conflict of interest: The authors state no conflict of interest.

Data availability statement: The data sets are created during the evolution of the network.

References

[1] Hamami L, Nassereddine B. A study of the main factors affecting wireless sensor networks. In2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC). IEEE; 2019 Dec. p. 211–215.
[2] Lanzolla A, Spadavecchia M. Wireless sensor networks for environmental monitoring. Sensors. 2021 Feb;21(4):1172.
[3] Mohapatra H, Rath AK. Fault-tolerant mechanism for wireless sensor network. IET Wireless Sens Syst. 2020 Feb;10(1):23–30.
[4] Xue X, Shanmugam R, Palanisamy S, Khalaf OI, Selvaraj D, Abdulsahib GM. A hybrid cross layer with harris-hawk-optimization-based efficient routing for wireless sensor networks. Symmetry. 2023 Feb;15(2):438.
[5] Kareem AT, Alrawi MA, Ali Israa T. Smart inventory control system based on wireless sensor network. Int J Eng Res Appl. 2017;7(8):40–7, (www.ijera.com).
[6] Asia AK, Fasli M. Deploying self-organisation to improve task execution in a multi-agent systems. In2017 3rd IEEE International Conference on Cybernetics (CYBCONF). IEEE; 2017 Jun. p. 1–8.
[7] Islam M, Kumar A, Hossain A. Study of wireless sensor network. Int J Sens Sens Network. 2019 Aug; 7:9–15.
[8] Al-Shebanee D. Ultra-low-power CMOS ring oscillator with minimum power consumption of 2.9 pW using low-voltage biasing technique. Open Eng. 2024;14(1):20220448.
[9] Loga-Księska W, Sordyl J, Ryguła A. Long-term urban traffic monitoring based on wireless multi-sensor network. Open Eng. 2020;10(1):197–208.
[10] Alzubaidi WK, Shaker SH. Secure routing scheme for clustered wireless sensor network (WSN). Interciencia J. 2018;43(10):48–61.
[11] Mosavvar H, Ghaffari A. Detecting faulty nodes in wireless sensor networks using harmony search algorithm. Wireless Pers Commun. 2018 Dec;103(4):2927–45.
[12] Loganathan S, Arumugam J, Chinnababu V. An energy-efficient clustering algorithm with self-diagnosis data fault detection and prediction for wireless sensor networks. Concurr Comput: Pract Exp. 2021 Sep;33(17):e6288.
[13] Gutiérrez S, Ponce H. An intelligent failure detection on a wireless sensor network for indoor climate conditions. Sensors. 2019 Feb;19(4):854.
[14] Bista R, Chaudhary M. A new fault detection approach in wireless sensor networks. In2022 14th International Conference on Software, Knowledge, Information Management and Applications (SKIMA). IEEE; 2022 Dec. p. 187–91.
[15] Palanikumar R, Ramasamy K. Effective failure nodes detection using matrix calculus algorithm in wireless sensor networks. Clust Comput. 2019 Sep;22(Suppl 5):12127–36.
[16] Zidi S, Moulahi T, Alaya B. Fault detection in wireless sensor networks through SVM classifier. IEEE Sens J. 2017 Nov;18(1):340–7.
[17] Jia S, Ma L, Qin D. Fault detection modelling and analysis in a wireless sensor network. J Sens. 2018;2018(1):7935802.
[18] Satyanarayana P, Mahalakshmi T, Sivakami R, Alahmari SA, Rajeyyagari S, Asadi S. A new algorithm for detection of nodes failures and enhancement of network coverage and energy usage in wireless sensor networks. Mater Today: Proc. 2023 Jan;80:1717–22.
[19] Wu X, Zhang X, Yuan S, Ji W. Fault detection method of wireless sensor network by using improved self-organizing map and trend correlation detection. In 2022 4th International Conference on Communications, Information System and Computer Engineering (CISCE). IEEE; 2022 May. p. 28–31.
[20] Umamaheswari S, Antony WS. Detection and correction of node failures in wireless sensor networks. In 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS). Vol. 1, IEEE; 2021 Mar. p. 1479–83.
[21] Shigaki S, Kuze N, Kominami D, Kashima K, Murata M. Self-organizing wireless sensor networks based on biological collective decision making for treating information uncertainty. In 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). IEEE; 2017 Oct. p. 167–74.
[22] Al-Karkhi A. Task Recovery in Self-Organised Multi-Agent Systems for Distributed Domains. Doctoral dissertation. Colchester, United Kingdom: University of Essex; 2018.
[23] Yu WT, Choi JW, Kim Y, Lee WH, Kim SC. Self-organizing localization with adaptive weights for wireless sensor networks. IEEE Sens J. 2018 Aug;18(20):8484–92.
[24] Balzano W, Stranieri S. A self-organization technique in wireless sensor networks to address node crashes problem and guarantee network connectivity. InWeb, Artificial Intelligence and Network Applications: Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA-2019). Springer International Publishing, vol. 33; 2019. p. 841–50.