Volume XVI Number 1, 2024

Optimizing IoT Data Aggregation: Hybrid Firefly-Artificial Bee Colony Algorithm for Enhanced Efficiency in Agriculture

Narayanaswamy Venkateswaran[1], Kayala Kiran Kumar[2], Kirubakaran Maheswari[3], Radha Vijaya Kumar Reddy[4], Sampath Boopathi[5]

[1] Department of Master of Business Administration, Panimalar Engineering College, Chennai, Tamil Nadu, India

[2] Department of F. E, Prasad V Potluri Siddhartha Institute of Technology, Vijayawada, Andhra Pradesh, India

[3] Department of Computer Science and Engineering, CMR Technical Campus, Hyderabad, Telangana, India

[4] Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India

[5] Department of Mechanical Engineering, Muthayammal Engineering College, Rasipuram, Namakkal District, Tamil Nadu, India

Abstract

The data aggregation process in this study has been enhanced by the hybrid firefly-artificial bee colony algorithm (HFABC) by increasing the average packet delivery ratio, end-to-end delay, and lifespan computation. In this study, HFABC and Multi Hop LEACH are two algorithms that are used to aggregate IoT data. Their performance is compared using evaluation criteria including average End-to-End Delay, PDR, and network lifetime. The HFABC method reduces average End-to-End Delay more effectively than Multi Hop LEACH, with gains of 2.20 to 8.66 %. This demonstrates how well it works to reduce the lag times for data transfer in IoT networks. With improvements ranging from 3.45% to 45.39%, HFABC has a greater success rate than Multi Hop LEACH in effectively delivering packets. HFABC increases network lifetime by 0.047 to 2.286 percent, indicating that it helps keep IoT networks operating for longer. For effective data aggregation in IoT networks, HFABC is a superior solution that decreases delays, improves packet delivery, and lengthens network lifetime.

Keywords

AHFABC, IoT, performance, Average End-To-End Delay, Average Packet Delivery Ratio, lifetime computation.

Venkateswaran, N., Kiran Kumar, K., Maheswari, K., Vijaya Kumar Reddy, R. and Boopathi, S. (2024) "Optimizing IoT Data Aggregation: Hybrid Firefly-Artificial Bee Colony Algorithm for Enhanced Efficiency in Agriculture", *AGRIS on-line Papers in Economics and Informatics*, Vol. 16, No. 1, pp. 117-130. ISSN 1804-1930. DOI 10.7160/aol.2024.160110.

Introduction

Agriculture is crucial for human sustenance, and IoT sensors enable farmers to monitor, manage, and optimize activities. Data-driven decisions and improved irrigation and fertilizer application improve crop yield and resource management. Traditional data aggregation methods in agriculture are time-consuming, labour-intensive, and inefficient. IoT technology offers automated data collection, real-time monitoring, and remote access, enabling continuous data collection from sensors and drones for precision agriculture practices (Patel et al., 2012). The network is divided into clusters and each cluster has a cluster head or coordinator responsible for data aggregation. Sensors or nodes within a cluster transmit their data to the cluster head, who aggregates it and transmits it to the base station. These clusters include LEACH (Low-Energy Adaptive Clustering Hierarchy), HEED (Hybrid Energy-Efficient Distributed clustering), PEGASIS (Power-Efficient Gathering in Sensor Information Systems) (Sinha and Lobiyal, 2013). Clustering is crucial in IoT applications for efficient data acquisition, dissemination, and network longevity, minimizing communication overhead and optimizing access to devices (Kumar and Zaveri, 2018). IoT networks consist of IP-enabled devices and sensor devices. Clustering designates a node as the Cluster Head (CH), providing scalability, reduced routing overhead, efficient bandwidth utilization, and improved stability. CHs minimize packet exchange, simplify network topology, and aggregate sensor data, reducing packet exchange rates (Sholla et al., 2017). Cluster Head optimizes device battery life using advanced scheduling algorithms, enabling low-power mode operation when inactive, avoiding collisions, and prolonging device lifespan.

LEACH is a routing protocol that balances energy consumption by randomly selecting cluster heads (CHs) for data collection and merging. It operates in two stages: setup phase, based on probabilistic models, and steady phase, optimizing energy usage and network lifetime (Zheng et al., 2017). TL-LEACH is an improved LEACH algorithm with a two-level clustering hierarchy, primary and secondary Cluster Heads, and simple sensing nodes. It uses data fusion and TDMA-based scheduling for efficient communication, ensuring fairness in CH selection and distribution (Boyinbode et al., 2010). FLOC is a distributed technique that creates well-balanced clusters with minimal overlap in wireless networks. It classifies radio nodes based on their proximity to the inner Cluster Head and uses an i-band membership preference. This enhances intra-cluster communication robustness, efficiency, and reduces interference, improving network performance and reliability (Abbasi and Younis, 2007). HEED optimizes energy efficiency in WirelessSensor Networks (WSNs) by considering communication cost and energy factors. DEEC is a multilevel algorithm for heterogeneous WSNs, ensuring low probability of selecting the same CH, adapting to energy consumption, and allowing adjustable CH selection probabilities (Katiyar et al., 2010). EECS involves dynamic cluster sizing based on distance from the base station, ensuring equal energy distribution among nodes. This approach improves network lifetime, connectivity, and reliable sensing capabilities, enhancing energy efficiency and performance in wireless networks (Katiyar et al., 2011).

EEUC divides nodes into clusters, reducing energy consumption for data forwarding. SEP improves LEACH by considering node heterogeneity and assigning higher probability of CH to advanced nodes, resulting in more efficient and balanced clustering (Tewari and Vaisla, 2014). The Deterministic Energy Efficient Clustering Algorithm (DEC) is a deterministic clustering protocol that minimizes uncertainties in Cluster Head selection. SECA ensures uniform cluster formation and load distribution by selecting CH nodes based on distance, and a centralized controller sink updates nodes for cluster formation (Rajput and Kumaravelu, 2018). NCACM is a centralized protocol for clustering heterogeneous wireless sensor networks to improve energy efficiency. It enhances the selection process of Cluster Head Nodes (CHNs) and considers factors like distance and battery power. NCACM maximizes overall network energy efficiency and performance by considering multiple parameters.

LEACH is a crucial hierarchical routing protocol in Wireless Sensor Networks (WSNs) to extend network lifetime. It uses cluster heads to collect and aggregate data from non-CH nodes, ensuring fair energy distribution and prolonging the network's operational lifespan (Lee et al., 2017). The ODL-CNN system uses IoT cameras for surveillance, utilizing IEHO algorithm optimization. Deep learning analysis generates sketches, and effectiveness is evaluated through examination and simulation, ensuring reliability and accuracy in fire surveillance applications (Elhoseny et al., 2020). The proposed approach improves traditional IDS performance and adaptability by combining MOPSO and Lévy flight randomization, introducing evolutionary and intelligent elements for effective multi-objective handling and enhanced intrusion detection capabilities (Hussein et al., 2022).

The PL50 Optimization method uses dynamic pheromone adjustment in real-time data for logistics optimization. Combining mathematical modeling, fuzzy time windows, and an improved Ant Colony algorithm, it enhances production logistics efficiency and effectiveness. Numerical experiments validate its viability and performance (Huang et al., 2020). The competitive nature and superior performance of MOPSO over traditional machine learning methods, NSGA-II, and filter-based methods highlight its effectiveness as a powerful optimization algorithm for solving multi-objective problems (Habib et al., 2020). The Ant Colony Optimization (ACO) multi-objective service selection algorithm delivers quick, effective solutions, improving recall and precision while maintaining high efficiency. This approach has potential for enhancing service selection processes in various applications (Tian et al., 2019). The modified Genetic Algorithm for Resource Selection (MGA-RS) algorithm identifies optimal data using a modified genetic algorithm, showing superiority when combined with kNN-based fitness function. This improves resource selection and classification accuracy in various applications (Bharti et al., 2019). Improved GA and DBN enhance intrusion attack recognition rates, reduce neural network complexity, and optimize IoT network performance for effective data transmission (Zhang et al., 2019). the Grey Wolf Algorithm-based intelligent approach for routing in IoT networks demonstrated its superiority over AFSA and ABC in terms of energy consumption and network throughput. The findings emphasize the effectiveness of utilizing GWA for optimizing routing decisions and improving the overall performance of IoT networks (Mahdy et al., 2023). Bee-Inspired Routing Algorithm (BIRA) is an energy-aware routing algorithm inspired by bee communication, designed for D2D IoT communication. It offers better packet delivery ratio, reduced energy consumption, and lower end-to-end delay, enhancing efficiency and reliability in IoT networks (Almazmoomi and Monowar, 2019). The Firefly algorithm improves mobile sink path optimization for sensor networks by optimizing storage and computing resources, improving connectivity and communication efficiency. It also shows superior performance in cluster head energy balance and network reliability compared to alternative methods (Sun et al., 2022). QoPF uses the Backtracking Search Optimization Algorithm (BSOA) to optimize resource allocation and meet QoS requirements in service-oriented IoT environments. It outperforms other techniques and enhances user experience by addressing metrics like jitter, delay time, and throughput (Badawy et al., 2020).

A routing scheme developed using CSA Cuckoo Search Algorithm demonstrated effectiveness in network routing, with promising results compared to other contemporary algorithms (Nagavalli and Ramachandran, 2019). Adaptive Immune algorithm and Endocrine regulation offer innovative solution for dynamic service issues, with simulation-based evaluation proving effectiveness and superiority over existing optimization techniques (Yang et al., 2019). A novel routing algorithm based on ACO optimizes path selection in IoT systems, addressing challenges in intersected areas. NS-2 evaluation shows effectiveness in improving energy consumption, end-to-end delay, packet loss ratio, bandwidth consumption, throughput, and control bit overheads (Said, 2017).

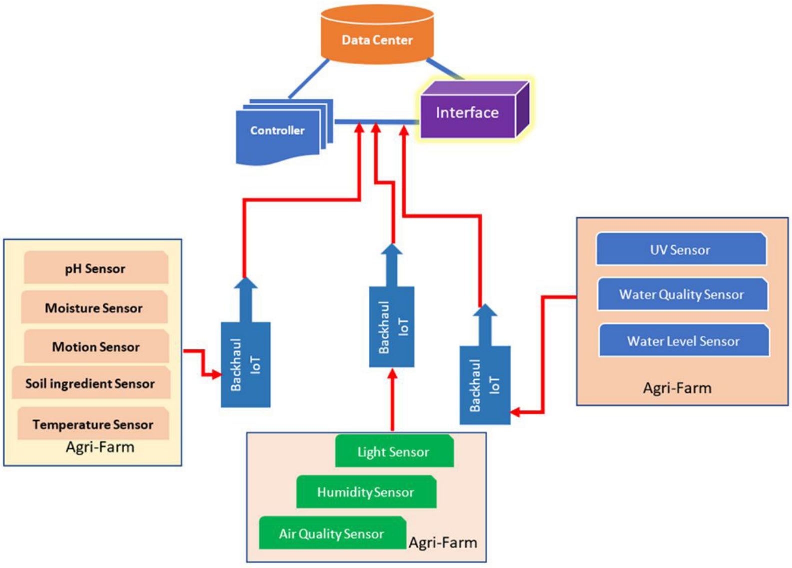
Thus, Advanced Clustering Algorithms are essential for IoT networks to address scalability, energy efficiency, and heterogeneity. Dynamic and adaptive algorithms adapt to network changes, while hybrid clustering and optimization techniques integrate. QoS-Aware Clustering optimizes energy efficiency and application-specific requirements, while Security and Privacy in Clustering addresses unique security challenges.

Materials and methods

In a low-power network topology designed for measuring and monitoring factors in a smart agricultural farm, the system includes (Figure 1):

- **IoT Sensor Nodes:** These nodes are responsible for collecting data from the farming environment, such as soil moisture, air humidity, temperature, nutrient ingredients of soil, pest images, and water quality. They transmit the collected data to the IoT backhaul devices.
- **Reduced-Function Devices (RFDs):** Some IoT sensor nodes are installed as RFDs, which have limited functionality and can only communicate with Full-Function Devices (FFDs). RFDs are designed to conserve energy and reduce investment costs. They cannot communicate with other RFDs.
- **IoT Backhaul Nodes:** IoT backhaul nodes not only function as IoT sensor nodes but also act as intermediate nodes in the network. They receive information from other IoT nodes and transmit it to the control center or central server. IoT backhaul nodes are typically installed as FFD devices, capable of connecting to both FFD and RFD devices.
- **Full-Function Devices (FFDs):** FFDs are devices that have complete functionality and can communicate with both FFD and RFD devices. They play a key role in relaying data between different nodes in the network, including IoT sensor nodes and IoT backhaul nodes.

The IoT sensor nodes, whether RFDs or FFDs, collect data from the farming environment and transmit it to the IoT backhaul nodes. The IoT backhaul nodes, particularly the FFDs, receive information from various nodes in the network and transmit it to the control center or central server for further processing and analysis



Source: Own processing

Figure 1: Low-power network topology: Smart agricultural farm.

(Cheng et al., 2022). This network topology enables efficient data collection, communication, and management in a smart farm environment.

The simulation parameters for the experiment include a total area of a sensor network deployment of 2000×2000 square meters, with the number of sensor nodes ranging from 100 to 1000, which determine the density and scale of the sensor network. The initial energy level (2 J) assigned to each sensor node in the network is used for sensing, communication, and other tasks.

The energy consumption rate (EC-energy per bit) for the electronic components of a sensor and electronic node during communication can be measured by a formula (Sharp et al., 2020).

$$E_C = E_{tx} + E_{rx} = (Eelec_{tx} + Eelec_{rx}) \times Tb = \\ = (Eelec_{tx} + Eelec_{rx}) \times L/R \quad (1)$$

where L is the packet length in bits, R is the data rate in bits per second, $Eelec$ - total energy consumption rate (energy per bit), E_{tx} - energy consumption rate for transmitting one bit of data, E_{rx} - energy consumption rate for receiving one bit of data, Tb -transmission time

The energy consumption rate (energy per bit per square meter) for free space transmission between sensor nodes can be calculated by the Equation 2.

$$E_{FC} = Eelec_{fs} * d^2 \quad (2)$$

where $Eelec_{fs}$ - energy consumption rate per bit for free space transmission, d is the distance between the transmitting and receiving nodes in meters.

The energy consumption rate (E_{mp}) for multi-path fading transmission calculated by the Equation 3. It can be measured by energy dissipation per bit of data ($Eelec_{mp}$) with respect to distance between the transmitting and receiving nodes (d) in meters.

$$E_{MC} = Eelec_{mp} \times d^4 \quad (3)$$

The reference distance (d_f) at which the E_{fs} energy model is calibrated is measured in meters and is used to determine the energy consumption for free space transmission. The maximum communication range between sensor nodes (d_{max})in the network can be measured in meters. Packet length measures the data packet transmitted between sensor nodes, indicating the maximum distance for direct communication.

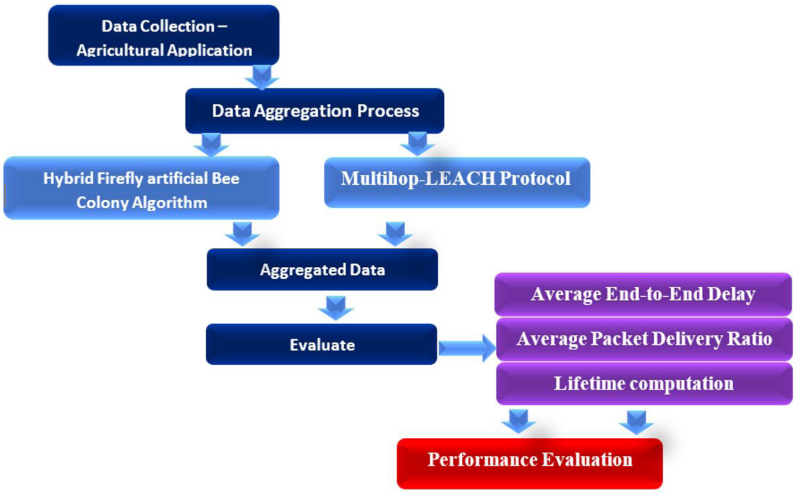
Table 1 displays IoT device parameters in agriculture, including coverage, sensor node count, initial energy consumption, communication distances, and packet length. Figure 2 shows IoT data aggregation performance analysis using Multihop Leach Protocol and HFABC.

The average End-to-End Delay and average Packet Delivery Ratio (PDR) are commonly used metrics to evaluate the performance of network communication. Here are the formulas

S. No.	Parameter	Value
1	Area	2000 * 2000 m^2
2	Number of Sensor Nodes	100,200,300,400,500,600,700,800,900, 1000
3	Initial Energy of Nodes	2 Jules
4	$E_{consume}$	50 nJ/bit
5	E_{FC}	10 pJ/bit/m^2
6	E_{MC}	0.0013pJ/bit/m^4
7	d_f	87
8	d_{max}	25 m
9	Packet length	30 bit

Source: Own processing

Table 1: Smart agricultural farm IoT system energy parameters and their values.



Source: Own processing

Figure 2: Schematic arrangement for performance analysis for IoT data aggregation using Multihop Leach Protocol and HFABC.

for calculating these metrics (Khan et al., 2013):

Average End-to-End Delay: The average End-to-End Delay is the average time a packet takes to travel from source to destination in a network, calculated by summing individual packet delays and dividing by total packets using the Equation 4.

Average End-to-End Delay = (Sum of Delays for all packets) / (Total number of packets) (4)

Average Packet Delivery Ratio (PDR): It measures the efficiency and reliability of message delivery by comparing the number of successfully delivered packets to the total number sent within the network (Equation 5).

Average PDR = (Number of successfully delivered packets) / (Total number of packets sent) (5)

Lifetime computation: Estimate lifetime computation for IoT devices using the Equation 6.

Lifetime = (Initial Energy / Energy Consumption Rate) × (1 - (1 - *PDR*)^n) (6)

Where, *Initial Energy* - The initial energy level of the sensor nodes, *Energy Consumption Rate* - The rate at which the sensor nodes consume energy during their operations, *PDR* (Packet Delivery Ratio) - The ratio of successfully delivered packets to the total number of packets sent and *n* - The number of rounds or cycles.

Multihop-LEACH Protocol

The Multihop-LEACH Protocol is an extension of the LEACH protocol for multi-hop communication in Wireless Sensor Networks (WSNs). The Multihop-LEACH Protocol improves network scalability, energy efficiency, and connectivity by enabling data transmission through intermediate nodes. Clusters form, with a designated Cluster Head responsible for data aggregation and transmission. This approach extends coverage and enables efficient data routing over longer distances, enabling WSNs to monitor and collect data from a wider geographic area (Patil, 2012). Multi-hop LEACH improves network scalability, energy efficiency, and coverage area by optimizing data transmission over long distances (Biradar et al., 2011).

The implementation steps of the Multihop-LEACH Protocol are as follows:

- The network is initialized by deploying sensor nodes in the target area. Each node is equipped with energy and communication capabilities.
- Nodes organize themselves into clusters with a designated Cluster Head (CH). The CH selection process can be based on various factors such as energy level, distance to the base station, or a combination of these factors.
- Non-Cluster Head nodes collect data from their sensing environment and transmit it to the CH within their cluster.
- The CH collects the received data from its member nodes and performs data aggregation to reduce redundancy and minimize the amount of data to be transmitted further.
- Instead of directly transmitting data to the base station, the CHs utilize other CHs as relay nodes to establish multi-hop communication. The data is forwarded from one CH to another until it reaches the base station.
- To ensure balanced energy consumption, the role of CH is rotated among nodes in subsequent rounds. Nodes with higher energy levels are selected as CHs, distributing the energy load throughout the network.
- The exact formulas used in the Multihop-LEACH Protocol may vary depending on the specific implementation and variations of the protocol. However, some commonly used formulas in the protocol may include:
- The formula to determine the probability of a node becoming a Cluster Head, which can be based on parameters like residual energy, distance to the base station, or a combination of factors.
- A formula to calculate the optimal path for data transmission through intermediate CHs, considering factors such as distance, energy consumption, or link quality.

Hybrid firefly - artificial Bee colony algorithm (HFABC)

The hybrid firefly-artificial bee colony algorithm solves complex optimization problems by combining fireflies and artificial bees. Fireflies communicate using light signals, while artificial bees mimic honey bees' foraging behavior, using employed, onlooker, and scout bees to explore solutions and exploit promising regions. The hybrid firefly-artificial bee colony algorithm combines firefly's attractiveness with artificial bee colony's exploration and exploitation strategies, increasing diversity and efficiency. This adaptive optimization tool exploits local and global search spaces, making it popular in engineering, economics, and other fields (Mallala et al., 2022).

The algorithm randomly places fireflies in a search space and evaluates their objective function using artificial bees. Iteratively, iterates until a termination condition is met, encouraging exploration and exploitation. Onlooker bees select employed bees based on fitness, and employed bees generate new solutions using exploration and exploitation strategies. The algorithm evaluates and updates the best solution if a better one is found. An explanation of the different components and steps in the provided MATLAB code for the hybrid firefly-artificial bee colony algorithm (Mallala et al., 2022):

- **Initialization of parameters and variables:** Code initializes parameters like coverage area, sensor node number, energy consumption, communication distances, and packet length.
- **Main loop for different number of sensor nodes:** Loop iterates over sensor node values, executing and evaluating algorithm for specified numbers.
- **Firefly initialization:** Code generates random initial firefly positions in specified coverage area for each iteration.
- **Evaluation of firefly fitness:** The code evaluates firefly fitness using the evaluateFitness() function, solving a specific problem by calculating fitness based on provided parameters.
- **Artificial bee initialization:** The employed bees are initialized based on the positions of the fireflies.
- **Best solution initialization:** The code initializes the best solution and best fitness variables to track the best solution found during the algorithm execution.

- **Main algorithm loop:** The code enters a loop that iterates for a specified number of iterations. This loop represents the main algorithm execution.
- **Firefly movement:** The **moveFireflies()** function is called to update the positions of the fireflies based on their attractiveness and distance. The fireflies move towards brighter and closer fireflies, exploring and exploiting the search space.
- **Artificial bee movement:** The **moveBees()** function is called to update the positions of the employed and onlooker bees based on the positions of the fireflies. This movement allows the bees to explore and exploit the search space using the employed bees' generated solutions.
- **Evaluation of employed bees' fitness and updating best solution:** The fitness of the employed bees' positions is evaluated using the **evaluateFitness()** function. The code then updates the best solution and best fitness if a better fitness value is found among the employed bees' solutions.
- **Scout bees exploration:** The **scoutBeesExploration()** function is called to perform exploration for the employed bees. It updates their positions to explore new regions of the search space.
- **Output the best solution found:** After the main algorithm loop completes, the code outputs the best solution and its corresponding fitness for the current number of sensor nodes.

Hybrid firefly-artificial bee colony algorithm for IoT agriculture devices, enabling customization and adaptation (Table 2).

Parameter	Example Value
Population Size	50
Iterations/Generations	500
Attractiveness	0.7
Distance	Calculated using a distance metric (e.g., Euclidean or Manhattan)
Exploration Rate	0.6
Exploitation Rate	0.4
Fitness Function	Problem-specific
Initial Energy of Nodes	2 Jules
Energy Consumption Rate	50 nJ/bit
E_{FC}	10 pJ/bit/m²
E_{MC}	0.0013 pJ/bit/m⁴
Communication Range	25 meters

Source: Own processing

Table 2: Hybrid firefly and artificial bee colony algorithm.

Experiments with sensor nodes ranging from 100 to 1000 are conducted, with lifetime computation performed for 0 to 1000 rounds. Results include average End-to-End Delay, Packet Delivery Ratio, and Lifetime Computation. Network lifetime is evaluated based on alive edge sensor nodes, providing insights into performance changes as rounds progress. A MATLAB code program is provided to simulate average End-to-End Delay, Packet Delivery Ratio, and Lifetime computation for the hybrid firefly artificial bee colony algorithm (Yang and Slowik, 2020; Karaboga and Basturk, 2007; Karaboga and Basturk, 2008). MATLAB code simulates average end-to-end delay and average PDR for hybrid firefly-artificial bee colony algorithm using IoT device parameters in agriculture. The IoT device initializes parameters like area, sensor count, energy consumption, communication distances, and packet length. A hybrid firefly-artificial bee colony algorithm is implemented, considering population size, generation iteration, attractiveness, distance calculation, exploration, exploitation rates, and fitness function. The simulation loop iterates over 100-1000 sensor nodes and rounds for lifetime computation. The algorithm calculates average End-to-End Delay and Average PDR after each round, and displays results for each iteration.

- **Fitness Calculation (calculate Fitness function):** Fitness calculation function evaluates position fitness based on optimization goals and problems, determining position suitability.
- **Firefly Movement (firefly Movement function):** Firefly movement function adjusts position based on attractiveness and distance, aiding search space exploration.
- **Artificial Bee Movement (artificialBeeMovement function):** Artificial bee movement function simulates population movement based on exploration and exploitation rates.
- **Update Best Solution (updateBestSolution function):** Update best solution function compares fitness values for optimal performance in iterations.
- **Calculate End-to-End Delay and PDR (calculate EndToEndDelayAndPDR function):** Function calculates End-to-End Delay and Packet Delivery Ratio by determining optimal position, measuring packet travel time, and dividing successful packets.
- **Update Lifetime Computation (updateLifetime function):** Update lifetime computation function updates network lifetime based on PDR, node number,

initial energy, energy consumption rate, and message delivery reliability.

- **Simulation Results Storage:** Code simulates Hybrid Firefly-Artificial Bee Colony algorithm, evaluating metrics, and storing results.

Results and discussion

Table 3 and Figure 3 present the Average Packet Delivery Ratio (PDR) results for both the Multi Hop LEACH and HFABC algorithms. The "Number of Rounds" column indicates the rounds or iterations of the simulation, while the "Multi Hop LEACH" and "HFABC" columns indicate the PDR values achieved at a particular round. The "Percentage of Improvement" column shows the percentage improvement of HFABC over Multi Hop LEACH in terms of PDR (Cheng et al., 2022; Sharp et al., 2020; Khan et al., 2013).

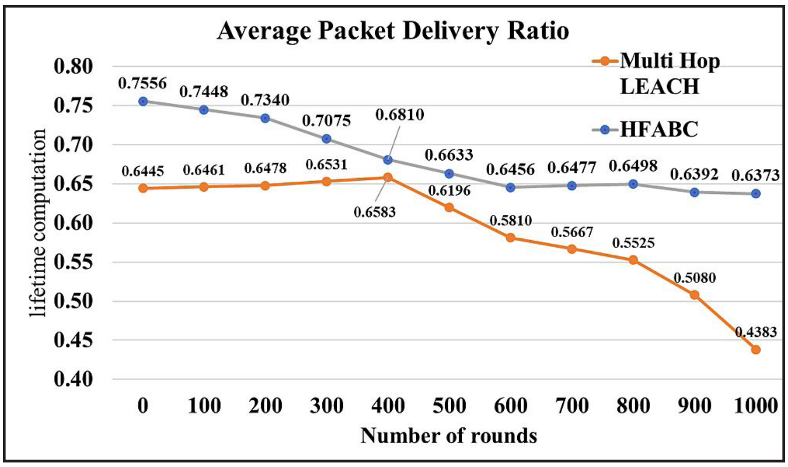
Average Packet Delivery Ratio: Comparative analysis

The PDR is an important metric that measures the ratio of successfully delivered packets to the total number of packets sent. At the beginning of the simulation, HFABC achieved a PDR of 0.7556, 17.24% higher than Multi Hop LEACH's PDR of 0.6445. This indicates that HFABC outperforms Multi Hop LEACH in terms of packet delivery, providing a higher percentage of successful packet transmissions. HFABC consistently achieves a higher PDR than Multi Hop LEACH, resulting in percentage improvements ranging from 3.45% to 45.39%. For example, at round 400, HFABC achieved a higher PDR of 0.6810, representing a 3.45% improvement over Multi Hop LEACH. As the rounds continue, the percentage improvement increases, indicating that HFABC consistently achieves a higher PDR than Multi Hop LEACH.

Number of Rounds	Multi Hop LEACH	HFABC	Percentage of improvement (%)
0	0.6445	0.7556	17.24
100	0.6461	0.7448	15.26
200	0.6478	0.7340	13.30
300	0.6531	0.7075	8.34
400	0.6583	0.6810	3.45
500	0.6196	0.6633	7.05
600	0.5810	0.6456	11.13
700	0.5667	0.6477	14.29
800	0.5525	0.6498	17.62
900	0.5080	0.6392	25.82
1000	0.4383	0.6373	45.39

Source: Own processing

Table 3: Average Packet Delivery Ratio.



Source: Own processing

Figure 3: The variation of Average Packet Delivery Ratio: Multi-Hop Leach and HFABC.

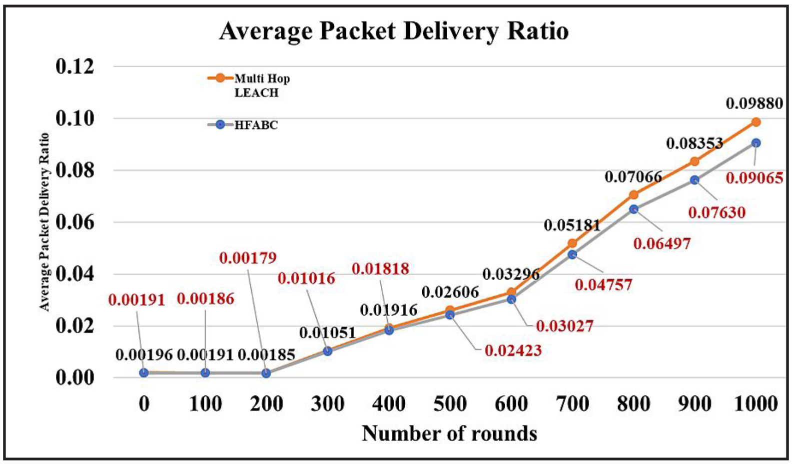
Multi Hop LEACH's PDR dropped significantly to 0.4383 by round 1000, while HFABC maintained a much higher PDR of 0.6373, resulting in a 45.39% improvement over Multi Hop LEACH. HFABC optimizes fireflies and artificial bees' movement to improve data transmission efficiency, resulting in higher packet delivery percentages. This enhances reliability and performance in IoT networks, supporting various agricultural applications and enhancing data transmission (Sharp et al., 2020; Cheng et al., 2022).

The optimum results of Table 3, which presents the Average Packet Delivery Ratio (PDR) for the HFABC algorithm compared to the Multi Hop LEACH algorithm, can be observed at the highest number of rounds (round 1000). At this point, HFABC achieves a PDR of 0.6373, while Multi Hop LEACH only achieves a PDR of 0.4383. HFABC significantly outperforms Multi Hop LEACH in packet delivery performance, with a high PDR value of 0.6373 indicating successful delivery of a large percentage of packets. HFABC algorithm is crucial for reliable IoT data transmission in agriculture, ensuring accurate collection for decision-making and monitoring, supporting smooth agricultural applications.

Average End-to-End Delay: Comparative analysis

Table 4 and Figure 4 show average End-to-End Delay for Multi Hop LEACH and HFABC algorithms, with HFABC showing a percentage improvement over Multi Hop LEACH. Number of rounds: simulation iterations; Multi Hop LEACH: average End-to-End Delay achieved for given number of rounds.

Percentage of improvement (%): The percentage improvement of HFABC over Multi Hop LEACH, calculated as ((*Multi Hop LEACH - HFABC) / Multi Hop LEACH*) * 100.



Source: Own processing

Figure 4: The variation of Average End-to-End Delay: Multi-Hop Leach and HFABC.

Number of Rounds	Multi Hop LEACH	HFABC	Percentage of improvement (%)
0	0.00196	0.00191	2.20
100	0.00191	0.00186	2.57
200	0.00185	0.00179	3.51
300	0.01051	0.01016	3.29
400	0.01916	0.01818	5.10
500	0.02606	0.02423	7.03
600	0.03296	0.03027	8.15
700	0.05181	0.04757	8.18
800	0.07066	0.06497	8.05
900	0.08353	0.07630	8.66
1000	0.09880	0.09065	8.25

Source: Own processing

Table 4: Average End-to-End Delay.

The Table 5 compares HFABC algorithm to Multi Hop LEACH in terms of average End-to-End Delay. HFABC consistently achieves lower delays for increasing rounds, with a percentage improvement column indicating superior performance. The numerical values represent average delays for different simulation rounds.

For 0 rounds, Multi Hop LEACH achieves an average End-to-End Delay of 0.00196, while HFABC achieves a slightly lower value of 0.00191. HFABC outperforms Multi Hop LEACH in reducing average End-to-End Delay by 2.20%. As rounds increase, both algorithms decrease, indicating increased efficiency in packet transmission, resulting in reduced delays.

Looking at specific data points, for example, when there are 500 rounds, Multi Hop LEACH achieves an average End-to-End Delay of 0.02606, while HFABC achieves a lower value of 0.02423. This indicates that HFABC performs significantly better in this case, with an improvement of 7.03% over Multi Hop LEACH. The HFABC algorithm consistently outperforms Multi Hop LEACH in average End-to-End Delay reduction, with percentage improvements ranging from 2.20% to 8.66%. The HFABC algorithm outperforms Multi Hop LEACH in optimizing network routing and transmission mechanisms, resulting in lower average End-to-End Delay. This improvement is due to its hybrid nature, combining firefly optimization and artificial bee colony algorithms for efficient packet routing and transmission solutions.

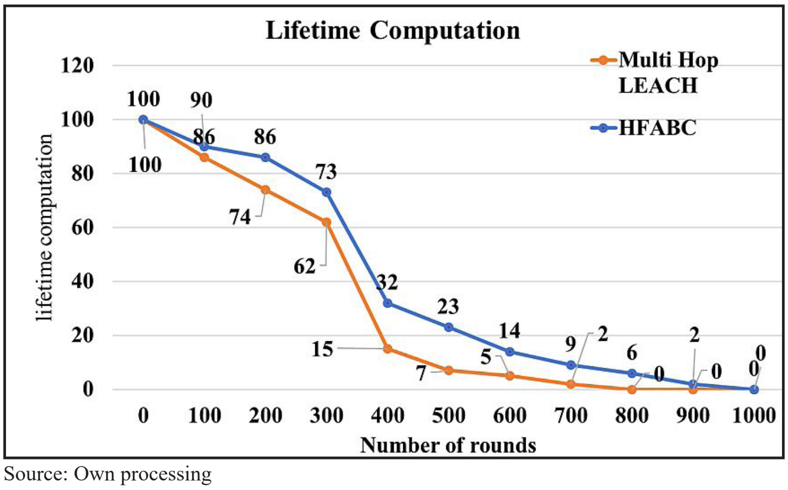
Lifetime computation: Comparative analysis

Table 5 and Figure 5 depict the lifetime computation results for the Hybrid Firefly-Artificial Bee Colony (HFABC) algorithm compared to the Multi Hop LEACH algorithm. The "Number of Rounds" column indicates the number of rounds or iterations

Number of Rounds	Multi Hop LEACH	HFABC	Percentage of improvement (%)
0	100	100	0
100	86	90	0.047
200	74	86	0.162
300	62	73	0.177
400	15	32	1.133
500	7	23	2.286
600	5	14	1.800
700	2	9	3.500
800	0	6	-
900	0	2	-
1000	0	0	-

Source: Own processing

Table 5. Lifetime computation for HFABC.



Source: Own processing

Figure 5: The variation of life time computation: Multi-Hop Leach and HFABC.

of the simulation, while the "Multi Hop LEACH" and "HFABC" columns represent the number of alive edge sensor nodes for each algorithm at a specific round. The "Percentage of Improvement" column shows the percentage improvement of HFABC over Multi Hop LEACH in terms of the number of alive nodes (Sun et al., 2022; Zhang et al., 2019).

HFABC had higher active edge sensor nodes than Multi Hop LEACH, indicating improved energy efficiency and longer network lifetime, with both algorithms having 100 active nodes. HFABC has a longer network lifetime due to its higher number of alive nodes. By round 400, HFABC had 32 active nodes, a 1.133% improvement. By round 500, it had 23 active nodes, a 2.286% improvement. HFABC consistently outperforms Multi Hop LEACH in terms of alive nodes, with 6 and 2 active nodes achieved at rounds 800 and 900, respectively. HFABC outperforms Multi Hop LEACH in terms of alive nodes. At rounds 800 and 900, HFABC achieves 6 and 2 alive nodes, respectively, while Multi Hop LEACH has no remaining nodes. The optimum result is observed at round 900, where HFABC achieves a network lifetime of 2.

Conclusions

According to the simulation results for average End-to-End Delay, average Packet Delivery Ratio (PDR), and Lifetime computation for the Multi Hop LEACH and HFABC algorithms over varying numbers of rounds.

- HFABC algorithm consistently achieves lower average End-to-End Delay values than Multi Hop LEACH, with a percentage improvement from 2.20% to 8.66%, demonstrating its superiority in reducing delays.
- HFABC outperforms Multi Hop LEACH in terms of Average Packet Delivery Ratio (PDR) for most rounds, ranging from 3.45% to 45.39%, indicating its ability to deliver more packets successfully.
- HFABC shows a significant improvement in lifetime compared to Multi Hop LEACH, with a percentage improvement ranging from 0.047% to 2.286% across rounds.

The HFABC algorithm outperforms Multi Hop LEACH in terms of average End-to-End Delay, Packet Delivery Ratio, and Lifetime computation. It achieves lower delays, higher PDR, and improved lifetime compared to Multi Hop LEACH. This highlights the effectiveness of the hybrid firefly-artificial bee colony algorithm in optimizing IoT data transmission in agricultural systems. HFABC improves overall performance and efficiency, leading to better decision-making and resource management, making it a promising choice for IoT applications in agriculture.

Corresponding author:
Sampath Boopathi
Department of Mechanical Engineering, Muthayammal Engineering College
Rasipuram, Namakkal, Tamil Nadu, 637 408, India
E-mail: boopasangee@gmail.com

References

[1] Abbasi, A. A. and Younis, M. (2007) "A survey on clustering algorithms for wireless sensor networks", *Computer Communications*, Vol. 30, No. 14-15, pp. 2826-2841. ISSN 0140-3664. DOI 10.1016/j.comcom.2007.05.024.

[2] Almazmoomi, A., and Monowar, M. (2019) "BIRA: A bee inspired routing algorithm for device-to-device ommunication in the internet of things", *Proceedings - 2019 International Conference on Information Technology, ICIT 2019*, Vol. 10, No. 11, pp. 325-330. DOI 10.1109/ICIT48102.2019.00064.

[3] Badawy, M. M., Ali, Z. H. and Ali, H.A. (2020). "QoS provisioning framework for service-oriented internet of things (IoT)", *Cluster Computing*, Vol. 23, No. 2, pp. 575-591, ISSN 1573-7543. DOI 10.1007/s10586-019-02945-x.

[4] Bharti, A., Chawla, S. S., Kumar, S., Kaur, S., Soin, D., Jindal, N. and Garg, R. (2019) "Asymptomatic bacteriuria among the patients of type 2 diabetes mellitus", *Journal of Family Medicine and Primary Care*, Vol. 8, No. 2, p. 539, ISSN 2249-4863. E-ISSN 2278-7135, ISSN 2249-4863. DOI 10.4103/jfmpc.jfmpc_403_18.

[5] Biradar, R. V., Sawant, S. R., Mudholkar, R. R. and Patil, V. C. (2011) "Inter-intra cluster multihop-leach routing in self-organizing wireless sensor networks", *International Journal of Research and Reviews in Computer Science*, Vol. 2, No. 1, pp. 88-95. ISSN 2079-2557.

[6] Boyinbode, O., Le, H., Mbogho, A., Takizawa, M. and Poliah, R. (2010) "A survey on clustering algorithms for wireless sensor networks", *Proceedings - 13th International Conference on Network-Based Information Systems, NBiS 2010*, Vol. 1, No. 2-3, pp. 358-364. ISBN 9781424480531. DOI 10.1109/NBiS.2010.59.

[7] Cheng, Y., Zhao, H. and Xia, W. (2022) "Energy-Aware Offloading and Power Optimization in Full-Duplex Mobile Edge Computing-Enabled Cellular IoT Networks", *IEEE Sensors Journal*, Vol. 22, No. 24, pp. 24607-24618. ISSN 1558-1748.

[8] Elhoseny, M., Shankar, K., Lakshmanaprabu, S. K., Maseleno, A. and Arunkumar, N. (2020) "Hybrid optimization with cryptography encryption for medical image security in Internet of Things", *Neural Computing and Applications*, Vol. 32, No. 15, pp. 10979-10993, ISSN 1433-3058. DOI 10.1007/s00521-018-3801-x.

[9] Habib, N., McCabe, C., Medina, S., Varshavsky, M., Kitsberg, D., Dvir-Szternfeld, R., Green, G., Dionne, D., Nguyen, L., Marshall, J. L., Chen, F., Zhang, F., Kaplan, T., Regev, A. and Schwartz, M. (2020) "Disease-associated astrocytes in Alzheimer's disease and aging", *Nature Neuroscience*, Vol. 23, No. 6, pp. 701-706. ISSN 1546-1726. DOI 10.1038/s41593-020-0624-8.

[10] Huang, C., Wang, Y., Li, X., Ren, L., Zhao, J., Hu, Y., Zhang, L., Fan, G., Xu, J., Gu, X., Cheng, Z., Yu, T., Xia, J., Wei, Y., Wu, W., Xie, X., Yin, W. , Li, H., Liu, M., Xiao, Y., Gao, H., Guo, L., Xie, J., Wang, G., Jiang, R., Gao, Z., Jin, Q., Wang, J. and Cao, B. (2020) "Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China", *The Lancet*, Vol. 395, p. 10223, pp. 497-506. ISSN 1474-547X. DOI 10.1016/S0140-6736(20)30183-5.

[11] Hussein, H. M., Katzis, K., Mfupe, L. P. and Bekele, E. T. (2022) "Performance Optimization of High-Altitude Platform Wireless Communication Network Exploiting TVWS Spectrums Based on Modified PSO", *IEEE Open Journal of Vehicular Technology*, Vol. 3, pp. 356-366. ISSN 2644-1330. DOI 10.1109/OJVT.2022.3191762.

[12] Karaboga, D. and Basturk, B. (2007) "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm", *Journal of Global Optimization*, Vol. 39, pp. 459-471. ISSN 0925-5001. DOI 10.1007/s10898-007-9149-x.

[13] Karaboga, D. and Basturk, B. (2008) "On the performance of artificial bee colony (ABC) algorithm", *Applied Soft Computing*, Vol. 8, No. 1, pp. 687–697. ISSN 1568-4946. DOI 10.1016/j.asoc.2007.05.007.

[14] Katiyar, V., Chand, N., Gautam, G. C. and Kumar, A. (2011) "Improvement in LEACH protocol for large-scale wireless sensor networks", *2011 International Conference on Emerging Trends in Electrical and Computer Technology*, pp. 1070-1075. ISBN 9781424479252. DOI 10.1109/ICETECT.2011.5760277.

[15] Katiyar, V., Chand, N. and Soni, S. (2010) "Clustering algorithms for heterogeneous wireless sensor network: A survey", *International Journal of Applied Engineering Research*, Vol. 1, No. 2, p. 273. ISSN 0973-4562.

[16] Khan, M. F., Felemban, E. A., Qaisar, S. and Ali, S. (2013) "Performance analysis on packet delivery ratio and end-to-end delay of different network topologies in wireless sensor networks (WSNs)", *2013 IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks*,, p. 324-329. ISBN 9781479934300. DOI 10.1109/MSN.2013.74.

[17] Kumar, J. S. and Zaveri, M. A. (2018) "Clustering Approaches for Pragmatic Two-Layer IoT Architecture", *Wireless Communications and Mobile Computing*, Vol. 2018, p. 8739203. ISSN 1530-8677. DOI 10.1155/2018/8739203.

[18] Lee, J. Y., Jung, K. D., Moon, S. J. and Jeong, H. Y. (2017) "Improvement on LEACH protocol of a wide-area wireless sensor network", *Multimedia Tools and Applications*, Vol. 76, p. 19843-19860. ISSN 1573-7721. DOI 10.1007/s11042-016-3732-4.

[19] Mahdy, A., Shaheen, A., El-Sehiemy, R., Ginidi, A. and Al-Gahtani, S. F. (2023) "Single-and Multi-Objective Optimization Frameworks of Shape Design of Tubular Linear Synchronous Motor", *Energies*, Vol. 16, No. 5, p. 2409. ISSN 1996-1073. DOI 10.3390/en16052409.

[20] Mallala, B., Papana, V. P., Sangu, R., Palle, K. and Chinthalacheruvu, V. K. R. (2022) "Multi-objective optimal power flow solution using a non-dominated sorting hybrid fruit fly-based artificial bee colony", *Energies*, Vol. 15, No. 11, p. 4063. ISSN 1996-1073. DOI 10.3390/en15114063.

[21] Nagavalli, S., and Ramachandran, G. (2019) "An efficient routing scheme using cuckoo search algorithm forinternet of things", *Journal of Computational and Theoretical Nanoscience*, Vol. 16, No. 5-6, pp. 2045-2049. ISSN 15461963. DOI 10.1166/jctn.2019.7847.

[22] Patel, P. D., Lapsiwala, P. B. and Kshirsagar, R. V. (2012) "Data aggregation in wireless sensor network", *International Journal of Managment, IT and Engineering*, Vol. 2, No. 7, pp. 457-472. E-ISSN 2249-0558.

[23] Patil, V. P. (2012) "Efficient AODV Routing Protocol for MANET with enhanced packet delivery ratio and minimized end to end delay", *International journal of scientific and Research Publications*, Vol. 2, No. 8, pp. 1-6. E-ISSN 2250-3153.

[24] Rajput, A. and Kumaravelu V. B. (2018) "Clustering techniques of Wireless sensor networks for Internet of Things", *ARPN Journal of Engineering and Applied Sciences*, Vol. 13, No. 5, pp. 1715-1733. ISSN 18196608.

[25] Said, O. (2017) "Analysis, design and simulation of Internet of Things routing algorithm based on ant colony optimization", *International Journal of Communication Systems*, Vol. 30, No. 8, p. e3174. ISSN 1099-1131. DOI 10.1002/dac.3174.

[26] Sharp, T. R., Zhivov, A., Haase, M. and Rose, J. (2020) "Energy master planning: identifying framing constraints that scope your technology options", *ASHRAE Transactions*, Vol. 126, No. 1. ISSN 0001-2505.

[27] Sholla, S., Kaur, S., Begh, G. R., Mir, R. N. and Chishti, M. A. (2017) "Clustering Internet of Things: A Review", *Journal of Science and Technology: Issue on Information and Communications Technology*, Vol. 3, No. 2, 21 p., ISSN 1859-1531. DOI 10.31130/jst.2017.61.

[28] Sinha, A. and Lobiyal, D. K. (2013) "Performance evaluation of data aggregation for cluster-based wireless sensor network", *Human-Centric Computing and Information Sciences*, Vol. 3, pp. 1-17. E-ISSN 2192-1962. DOI 10.1186/2192-1962-3-13.

[29] Sun, X., Xu, N., Yao, M., Cai, F. and Wu, M. (2022) "Efficient feedback linearization control for an IPMSM of EVs based on improved firefly algorithm", *ISA Transactions*, Vol. 134, pp. 431-441. ISSN 0019-0578. DOI 10.1016/j.isatra.2022.08.013.

[30] Tewari, M. and Vaisla, K. S. (2014) "Performance study of SEP and DEC hierarchical clustering algorithm for heterogeneous WSN", *Proceedings - 2014 6th International Conference on Computational Intelligence and Communication Networks, CICN 2014*, p. 385-389. ISBN 9781479950775. DOI 10.1109/CICN.2014.92.

[31] Tian, S., Wang, T., Zhang, L. and Wu, X. (2019) "An Energy-Efficient Scheduling Approach for Flexible Job Shop Problem in an Internet of Manufacturing Things Environment", *IEEE Access*, Vol. 7, pp. 62695-62704. ISSN 2169-3536. DOI 10.1109/ACCESS.2019.2915948.

[32] Yang, X.-S. and Slowik, A. (2020) "Firefly algorithm", a chapter In: *Swarm Intelligence Algorithms*, 1st ed., CRC Press, p. 163-174, eBook. ISBN 9780429422614. DOI 10.1201/9780429422614-13.

[33] Yang, Y., Zhang, S., Zhang, X., Gao, L., Wei, Y. and Ji, Y. (2019) "Detecting topology freezing transition temperature of vitrimers by AIE luminogens", *Nature Communications*, Vol. 10, No. 1, p. 3165. ISSN 2041-1723. DOI 10.1038/s41467-019-11144-6.

[34] Zhang, Y., Li, P. and Wang, X. (2019) "Intrusion Detection for IoT Based on Improved Genetic Algorithm and Deep Belief Network", *IEEE Access*, Vol. 7, pp. 31711-31722. ISSN 2169-3536. DOI 10.1109/ACCESS.2019.2903723.

[35] Zheng, X., Gao, W., Zhang, X., He, M., Lin, X., Cao, H., Zhang, Y. and Sun, Z. (2017) "Spent lithium-ion battery recycling – Reductive ammonia leaching of metals from cathode scrap by sodium sulphite", *Waste Management*, Vol. 60, pp. 680-688. ISSN 1879-2456. DOI 10.1016/j.wasman.2016.12.007.