Copyright ⓒ 2021 KSII

Mobile Sink Path Planning in Heterogeneous IoT Sensors: a Salp Swarm Algorithm Scheme

Ranida Hamidouche[1,2*], Zibouda Aliouat[1], Ado Adamou Abba Ari[2,3], and Abdelhak Gueroui[2]
[1] LRSD Laboratory, University of Ferhat Abbas Setif 1
Computer Science Department, El Bez, Setif , Algeria
[email: ranida.hamidouche@univ-setif.dz]
[email: zaliouat@univ-setif.dz]
[2] LI-PaRAD Laboratory, University of Versailles
Saint-Quentin-en-Yvelines, Versailles, France
[e-mail: mourad.gueroui@uvsq.fr]
[3] LaRI Laboratory, University of Maroua
Maroua, Cameroon
[e-mail: adoadamou.abbaari@gmail.com]
[*]Corresponding author: Ranida Hamidouche

Received May 31, 2020; revised February 11, 2021; revised February 8, 2021; accepted March 13, 2021; published June 30, 2021

Abstract

To assist in data collection, the use of a mobile sink has been widely suggested in the literature. Due to the limited sensor node’s storage capacity, this manner to collect data induces huge latencies and drop packets. Their buffers will be overloaded and lead to network congestion. Recently, a new bio-inspired optimization algorithm appeared. Researchers were inspired by the swarming mechanism of salps and thus creating what is called the Salp Swarm Algorithm (SSA). This paper improves the sink mobility to enhance energy dissipation, throughput, and convergence speed by imitating the salp’s movement. The new approach, named the Mobile Sink based on Modified Salp Swarm Algorithm (MSSA), is approved in a heterogeneous Wireless Sensor Network (WSN) data collection. The performance of the MSSA protocol is assessed using several iterations. Results demonstrate that our proposal surpass other literature algorithms in terms of lifespan and throughput.

Keywords: Sink Mobility, Salp Swarm Algorithm, Swarm Intelligence, Bio-inspiration, Heterogeneous WSN, IoT

http://doi.org/10.3837/tiis.2021.06.016 **ISSN : 1976-7277**

1. Introduction

Information gathering from heterogeneous WSNs utilizing mobile sinks has been a significant research theme this last few years [1]. The mobile sink directly collects data from Cluster Heads (CHs) through short-range communications by moving within the network without using routing [2]. This strategy offers an inherent load balancing by providing better energy efficiency in the network and alleviating hot spot issues [2, 3]. Yet, a slower mobile sink may cause a high latency in data gathering [3]. This can also cause data loss at the CHs level if their buffer is full. Thus, minimizing the data loss and enhancing the data gathering latency is a very difficult task to be reached in reality by the mobile sink [1]. Also, finding an optimal mobility trajectory for the mobile sink to achieve energy efficiency is a very hard task [2]. There exist certain time-sensitive applications such as medical surveillance where the slightest slowness in data delivery can cause irreparable damage. Accordingly, while data gathering, minimizing, or avoiding data delivery latency is a necessity. Accordingly, minimizing the buffer overflow problem may be reached by reducing data gathering latency of the mobile sink due to the optimal path planning strategy [1].

In Metaheuristics solutions have long proven their strength and efficiency in tackling the most encountered WSN issues. Nature-inspired algorithms take their inspiration from nature. Metaheuristics are subdivided into two main classes: swarm-based (e.g. bacterial colony and birds colony) and evolutionary-based (e.g. evolutionary and genetic algorithms). As already mentioned, swarm methods imitate the searching behavior of animals in nature like ant, bee, bat, bird, cuckoo, etc. These strategies demonstrated an important exhibition for data collection assignments utilizing sink mobility. The SSA algorithm has already been discussed in the mechanical design domain by Chen et al. [4]. The authors used quantum behavior particle swarm optimization to improve the original SSA version. However, Ateya et al. [5] analyzed SSA for a dynamic optimization algorithm with the introduction of chaotic maps in Software-defined networking (SDN). This roused us to look at the benefits of the recently recommended SSA [6] in the domain of WSN. The only contribution using the SSA algorithm in WSN that we could found is Huthaifa et al. [7]. The authors proposed a new scheme for node localization based on SSA. They aimed to reduce localization error, computing time, and the number of localized nodes. Past research has shown that swarm algorithms have handled WSN issues. In this paper, a modified SSA is utilized to improve the path taken by the mobile sink. The principal points of our work are as follows:

- We introduce a modified SSA by incorporating a new parameter α to adapt to the current best solution.
- It is the first contribution to be applied for sink mobility.
- The fitness function is formulated to minimize the data delivery latency and minimize the buffer overflow problem at the CHs level.

The rest of the paper is organized as follows: In section II related works about the data gathering using mobile sink-based swarm intelligence are described. The design of the modified version of the salp swarm algorithm and its hypothetical investigation are displayed in section III. Moreover, in section IV, the proposed solution is explained and validated by simulation results. Section V closes the proposed work.

2. Related Work

Sink mobility accomplishes an important role in WSN data collection. Recently, swarm intelligence demonstrated a valuable performance for sink mobility tasks. Jiang et al. [8] arranged the movement way of the mobile sink utilizing the Wolf Pack Algorithm (WPA). The authors determined the visiting point of each covering zone and decreased the computational effort required for voyaging way planning by the TSP algorithm. Their objective function was to minimize the length L(T) of a traveling-path constituted by several visited points T. The mobile sink visits each visiting point only once in each collection phase. Hamidouche et al. [9] presented a new pattern of Sink Mobility based on the Bacterial Foraging Optimization Algorithm (SMBFOA). The proposed protocol takes into account new characteristics inspired by the autonomous and intelligent movement of the Escherichia coli bacterium. Mechta and Harous [10] inspired by the Artificial Fish Swarm Algorithm (AFSA) for sink movement. Authors choose the best sink trajectory using organizational nature and searching the food behavior of fish. To reduce the distance traveled by the MDT during its tour, Yogarajan and Revathi [11] proposed a Discrete Firefly Algorithm for Mobile Data Transporter (DFAMDT) in a delay tolerant and low data rate WSNs. The proposed algorithm computes an optimal trajectory for the sink to visit the CHs to collect their data with minimal travel distance. Krishnan et al. [12] proposed a novel PSO based mobile data collector for information gathering. The network is subdivided into several groups. Chefs are chosen according to the highest residual energy. Then, the base station executes the PSO to find the shortest distance among all the chefs and pass this route information to the data collector. Yue et al. [13] analyzed the fault tolerance routing and established an intelligent fault-tolerant routing. The authors adopted an Artificial Bee Colony (ABC) swarm optimization to study the optimal recovery strategy of an alternate route. They used several factors to provide more accurate solutions and faster overall convergence performance for network optimization. Stephen and Mathivanan [14] combined PSO with a neural network to optimize sink motion. Wang et al. [2] proposed an energy-efficient Improved ACO-based approach to optimize the moving Mobile Sink's trajectory (IACO-MS). The authors considered the distance heuristic factor to enhance the global search ability, avoid being trapped in a local optimal solution, and improve the convergence rate. CHs are elected once to reduce the energy consumption of sensor nodes. Ren et al. [15] proposed an efficient data collection algorithm based on improved ABC algorithm to find the shortest traversal path. The authors used the initialization method of reverse learning and introduced the search equation inspired by the differential evolution algorithm.

Table 1 displays a portion of the prior research protocols for sink mobility. For all the more finding out about metaheuristics for sink portability, intrigued per-users are alluded to see the review paper [3].

3. Modified Salp Swarm Algorithm (MSSA) Approach for Sink Mobility

Similar to other swarm-based methods, salp's position is defined in a j-dimensional search space. Therefore, its value is stored in a two-dimensional matrix called X. There is also a food source called F in the search space as the swarm's target. The leader salp can update its position by using the next equation:

Table 1. A summary review of latest researches on the swarm algorithms for sink mobility

Authors	Mobility inspiration	Energy efficiency	Complexity	Link quality	Latency	Hetero.	Large-scale	Throughput	Topo.	Node's mobility	Connectivity	Reliability	Benchmark
Jiang et al. [8] (2019)	Wolf Pack Algorithm (WPA)	High	Low	High	Low	No	No	/	Plat	No	/	/	Random moving
Hamidouche et al. [9] (2018)	Bacterial Foraging Optimization Algorithm (BFOA)	High	Low	/	/	No	No	High	Cluster	No	/	/	CDCMP
Mechta and Harous [10] (2017)	Artificial Fish Swarm Algorithm (AFSA)	High	/	/	/	No	Yes	High	Cluster	No	/	/	LEACH-C
Yogarajan and Revathi [12] (2017)	Discrete firefly algorithm (DFA)	High	/	/	Low	No	Yes	/	Plat Cluster	No	/	/	ACO TSP OPT Span Hop Bounded Relay Spanning Itself
Krishnan et al. [13] (2018)	Particle Swarm Optimization (PSO)	High	/	/	/	No	Yes	/	Cluster	No	/	/	LEACH Genetic algorithm
Yue et al. [14] (2018)	Artificial Bee Colony Optimized Particle Swarm Optimization (ABC-PSO)	High	/	High	Low	No	No	/	Cluster	Yes	High	High	AODV-SMS AODV-SMS (PSO)
Stephen and Mathivanan [2] (2018)	Particle Swarm Optimization with Neural Network (PSO-NN)	High	/	/	Low	/	Yes	/	Cluster	/	/	/	Itself
Wang et al. [15] (2017)	Ant Colony Optimization (ACO)	High	/	/	Low	No	Yes	High	Cluster	No	/	/	ACO-M
Ren et al. [11] (2018)	Artificial Bee Colony optimization (ABC)	High	/	/	Low	No	No	High	Cluster	Yes	High	High	RW ACO

$$X_j^1 = \left\{ \begin{matrix} F_j + c_1\left(\left(ub_j - lb_j\right)c_2 + lb_j\right) if\, c_3 \geq 0 \\ F_j - c_1\left(\left(ub_j - lb_j\right)c_2 + lb_j\right) if\, c_3 < 0 \end{matrix} \right\} \tag{1}$$

where the meanings of all symbols are shown in **Table 2**.

Table 2. Meanings of symbols

Symbol	Meaning
X_j^1	leader position in the j^{th} dimension
F_j	food source position in jth dimension
ubj	upper bound of jth dimension
lbj	lower bound of jth dimension
c_1, c_2	random variables uniformly produced in the interval of [0,1].
c_3	random variable produced in the interval of [-∞, +∞].
l	current iteration
L	maximum number of iterations
X_j^i	position of i^{th} follower salp in j^{th} dimension
t	time
v_0	the initial speed

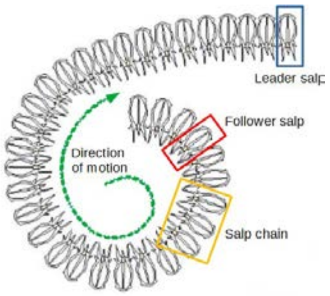


Fig. 1. Notion of leader and follower in salp's chain

$$c_1 = 2e^{-\left(\frac{4l}{L}\right)^2} \tag{2}$$

The coefficient c_1 is the crucial parameter in SSA. It gives harmony among exploitation and exploration capacities. The next equations are utilized to update the position of the followers:

$$X_j^i = \frac{1}{2}at^2 + v_0 \tag{3}$$

where $i \geq 2$; $a = \frac{v_{final}}{v_0}$ and $v = \frac{x - x_0}{t}$. Because in optimization the iteration is time, and considering $v_0 = 0$ and conflict between iterations is equal to 1, this equation can be refined as follows:

$$X_j^i = \frac{1}{2}\left(X_j^i + X_j^{i-1}\right) \tag{4}$$

As stated by Mirjalili et al. [6], the salp's position is changed using (1) and (4). In SSA, there is a solid relationship between the current best solution and the updated solution. A new parameter *α* [0.5, 1.5] is introduced into SSA to get the modified SSA. This new parameter quickens the speed of convergence during the exploration. It likewise makes a harmony among exploration and exploitation capacities to initially get away from a gigantic number of undesirable solutions and second to accomplish an exact recognition about the ideal solution.

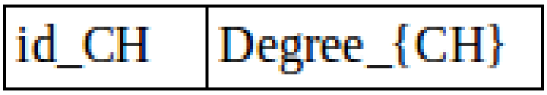
It is assumed that our network is subdivided into several clusters using Grey Wolf Optimizer (GWO) [16]. Our goal is that the sink collects data only from CHs by moving like a Salp. Since we use only one sink, which will represent the salp leader, we only need to update its position. The modified formula is stated by the next equations. If several data collectors are used in the network, the integration of the follower salps will be mandatory in this case. Each salp will represent a mobile sink.

$$X_{t+1}^1 = \begin{Bmatrix} \alpha \times F + c_1 \times c_2 \times X_t^1 if c_3 \geq 0.5 \\ \alpha \times F - c_1 \times c_2 \times X_t^1 if c_3 < 0.5 \end{Bmatrix} \tag{5}$$

$$Y_{t+1}^1 = \begin{Bmatrix} \alpha \times F + c_1 \times c_2 \times Y_t^1 if c_3 \geq 0.5 \\ \alpha \times F - c_1 \times c_2 \times Y_t^1 if c_3 < 0.5 \end{Bmatrix} \tag{6}$$

Besides, to improve data collection, the sink will first move to the CH with the largest number of members. To do so, after the clustering phase, CHs broadcast the message *INFO PACKET* containing their identities and the number of nodes in their cluster. The sink uses this information to update its coordinates. An example of this packet is shown in **Fig. 2**.

By choosing to go to the cluster having the largest number of members, i.e. moving to the CH having the smallest remaining memory space, the buffer overflow problem can be decreased. The parameter *F* will be represented by the maximum value of *Degree_{CH}* as formulated in (8), where $Degree_{CH}$ is the number of cluster member's calculated by (7), *Members* are the cluster member's and *M* is the number of sensors devised connected to a CH.



id_CH	Degree_{CH}

Fig. 2. Packet diffused by the CH after the end of the clustering phase.

$$Fitness = Degree_{CH} = \sum_{i=0}^{i=M}(Members) \tag{7}$$

$$F = max(Degree_{CH}) \tag{8}$$

Note that the computational complexity of the MSSA algorithm is calculated as follows.

Theorem. *The time complexity of the algorithm MSSA is O*(*n*)*, where n is the number of iterations.*

Proof. In the MSSA algorithm, the initialization phase takes *O*(1) time complexity. Calculating the fitness value of the mobile sink and the c_1 value takes *O*(1) time complexity. The position update of the leader salp x_1 iterates n times. Hence the total time complexity of MSSA is $O(n)$.

A detailed description of this algorithm is as follows.

Algorithm 1 Pseudo code of the MSSA algorithm.

```
- Initialize the position of the leader salp $x1$
  while $iteration < max\ number\ of\ iterations$ do
- Calculate the $fitness$ value of each cluster by Eq. 7
- $F$ = the best cluster to move to
- Update the parameter $c1$ by Eq. 2
- Update the position of $x1$ by Eq. 5 and 6
- iteration = iteration + 1
  end while
  return $x1$ position
```

4. Simulation Results and Discussions

In this section, through simulation experiments on the NS2 simulator, we evaluate the performance of the proposed MSSA. The network sensing area was assumed to be 100 × 100m^2 with 100 heterogeneous nodes randomly deployed.

The sink was initially positioned at (50, 50). The sensor node sends the collected data packets to its CH. After that, it is reforwarded to the Sink. The various parameters considered for simulations are given in **Table 3**. In this experiment, data collection is carried out based on the GWO clustering algorithm [16]. The MSSA is compared to itself using different values of the parameter α, a different number of iterations and different periods for data collection. Then, it is compared with the original SSA [6] to study the effects and the advantage that the α parameter adds to the algorithm. Finally, it is compared to the newest swarm algorithm named SMBFOA [9].

Table 3. Simulation parameters

Parameter	Value
Target area	100 × 100m^2
Sink Position	(50, 50)
Number of Nodes	100
Initial Energy of node	2J or 4J
Transmitter/Receiver Electronics - E_{elec}	50nJ/bit
Transmitter amplifier (free space) - ε_{fs}	100$pj/bit/m^2$
Transmitter amplifier (multipath) - ε_{mp}	0.013$pJ/bit/m^4$
Data Aggregation Energy cost – E_{DA}	50nJ/bit
Packet Size	8000$bits$
Sink collects data every	20s, 30s, 40s
Number of Iterations	20, 50, 100
α	∈ [0.5, 1.5]
c_1,c_2	∈ [0, 1]

Fig. 3 and **Fig. 4** represent the number of data packets delivered to the sink and the network lifespan using the MSSA algorithm collecting data at different times. It can be seen from the

curves of these figures that the longer the period for data collection is spaced, the larger of transmitted data packets are dropped and the larger of energy is consumed. This observation is since the sensor's buffers are overloaded, which generates a remarkable data loss. In this case, the nodes are forced to retransmit lost packets which cause more energy consumption and decrease the network lifespan. The more the sink passes frequently to collect the data of the nodes, the less will be the loss of the packets and energy. The best time to collect data is the one made every 20*s*.

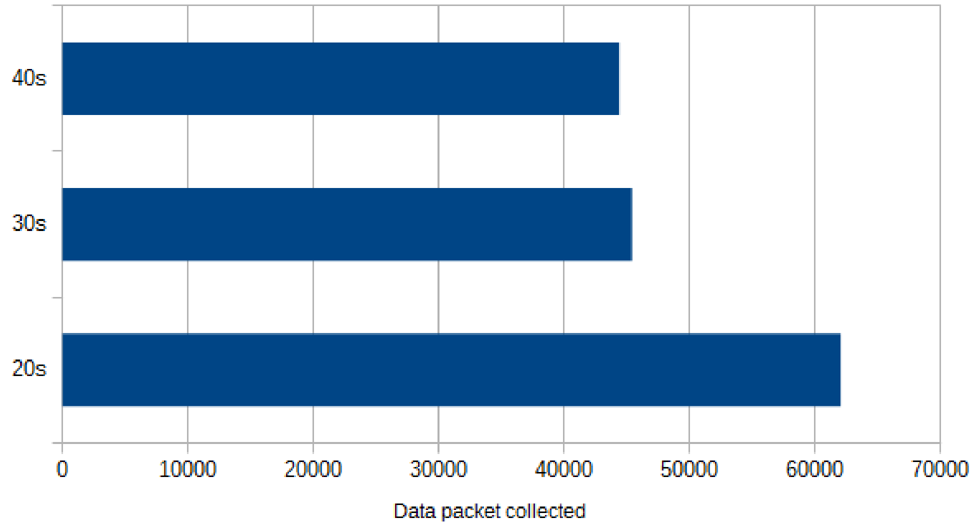


Fig. 3. Data packets collected in MSSA using different times for data collection

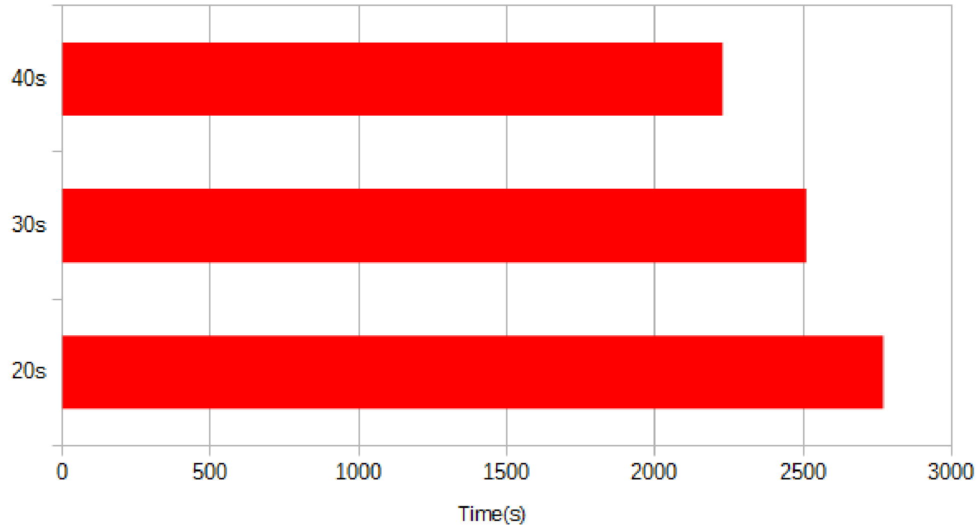


Fig. 4. MSSA Lifespan using different times for data collection

Fig. 5 and **Fig. 6** represent the number of data packets delivered to the sink and the network lifespan using the MSSA algorithm with a different number of iterations. The more iteration increases, the closer we get to the optimal solution. However, this negatively influences the network performance since the MSSA complexity is increased. For this reason, we notice that the number of packets and the lifespan decrease regarding the increase of iteration's number. In this case, it is necessary to find a compromise between the desired performances and the number of iterations.

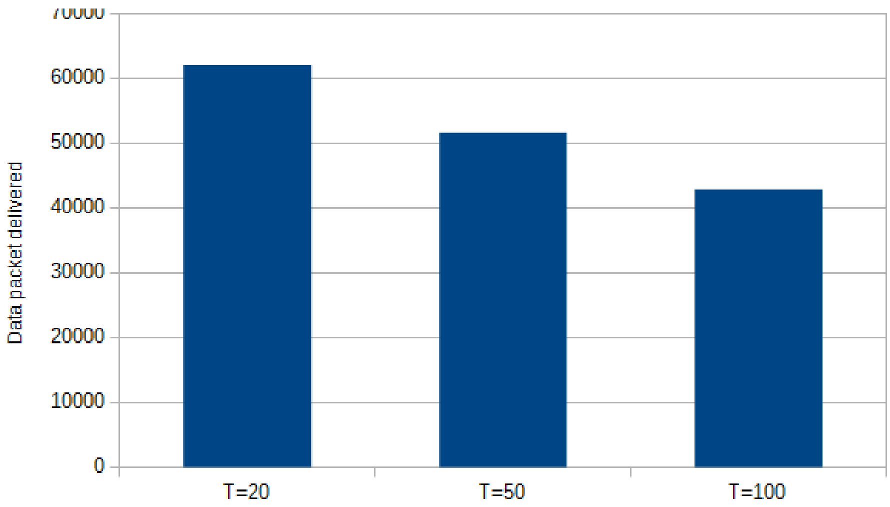


Fig. 5. Data packets collected in MSSA vs different iterations

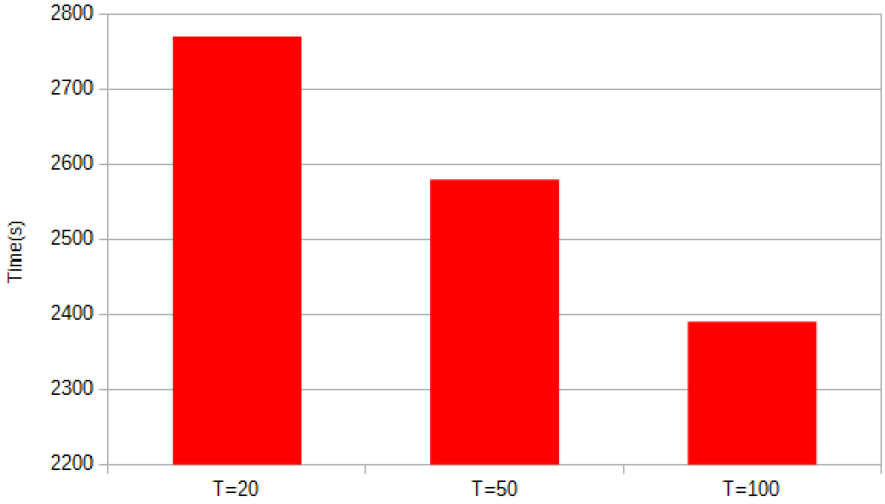


Fig. 6. MSSA Lifespan vs different iterations

For the rest of the simulation results, the mobile Sink moves at a constant speed and completes data collection for all CHs nodes every 20*s*. The parameters of the MSSA algorithm are as follows: the number of iterations is fixed to 20.

Fig. 7 represents the first node dead (FND), the middle node dead (MND), and the last node dead (LND) in the network using different values of the α parameter. According to the red curves, the first node died at t = 80*s* when α = 1.1. The network holds up until t = 140*s* when the value of α equals 0.6. From the yellow curves, half of the nodes are extinguished at t = 1000s when α = 0.9, against, it reaches up to t = 1800s when α = 0.6. By the green curves, the last operating node in the network died at t = 2450s when α = 1.3, while the network remains operational up to t = 2850s when α = 0.6. Based on these results, the value of the parameter α influences negatively or positively the network functionality. Therefore, it is necessary to choose its value according to consumer needs and the application necessities.

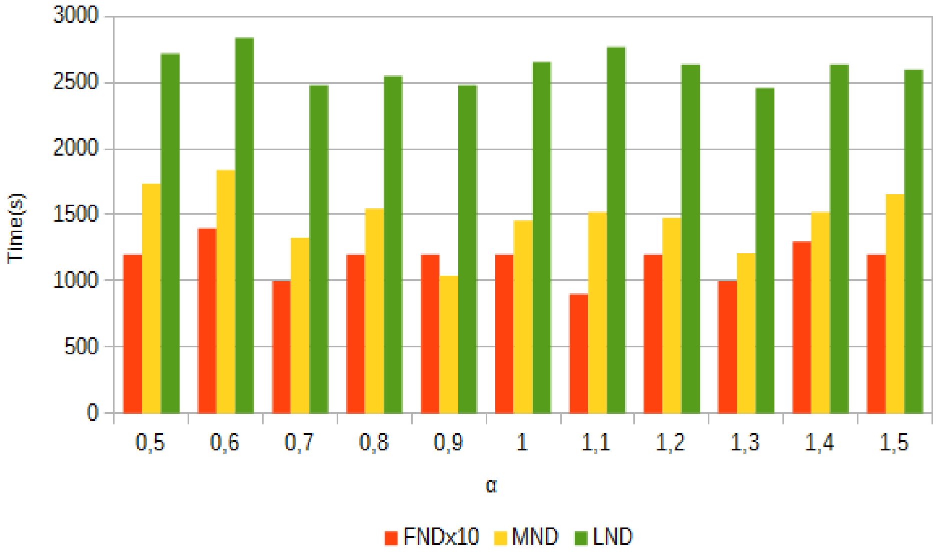


Fig. 7. FND, MND, and LND of MSSA algorithm vs α

Fig. 8 represents the network lifespan when using the MSSA algorithm with different values of the parameter α. The curve shows that the best lifetime is obtained when the value of α equals 0.5, 0.6, or 1.1. In hostile environments where charging or changing batteries is almost impossible, the energy factor is most considered when designing network protocols. As a result, the parameter α should be set to 0.5, 0.6, or 1.1 for a long operable network.

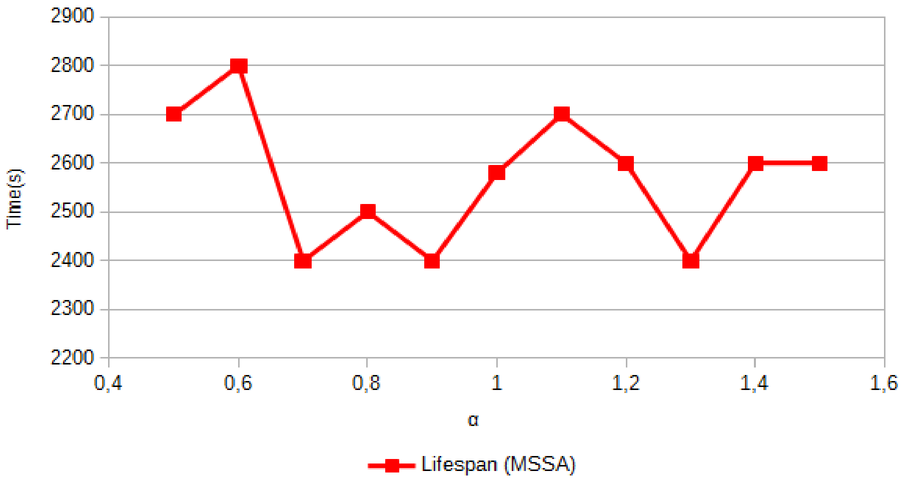


Fig. 8. MSSA lifespan vs α

Fig. 9 represents the number of data packets delivered to the sink using the MSSA algorithm with different values of the parameter α. We note that the best rate of delivered packets is obtained when $\alpha = 1.1$. In critical real-time applications where time constraint and urgency of data are paramount, it is better to set the value of α to 1.1.

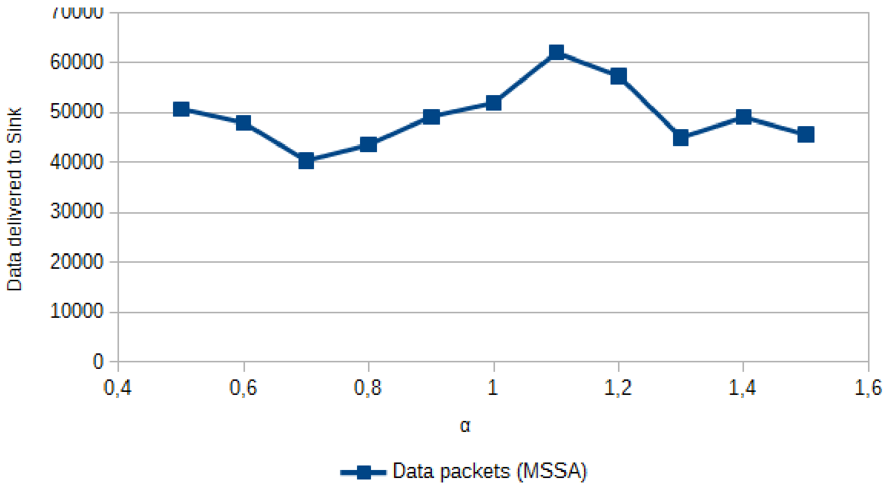


Fig. 9. MSSA data delivery vs α

Fig. 10 represents the distance separating the sink and the CHs using the three algorithms SMBFOA, SSA, and MSSA with $\alpha = 0.6$ and $\alpha = 1.1$. We note that the sink is closer to $3/5$ of CHs using MSSA when $\alpha = 0.6$, where only $2/5$ of CHs are closer to the sink when $\alpha = 1.1$. We notice also that, $3/5$ of CHs are closer to the sink using the MSSA algorithm comparing with its counterpart SMBFOA where only $2/5$ of the CHs are closer to the sink. Only one CH is farther from the sink when using the MSSA algorithm compared to the original SSA, where $4/5$ are farthest. MSSA algorithm finds the best coordinates for the sink compared to the SSA and SMBFOA algorithms which largely improves data delivery latency and makes it ideal for time-sensitive applications.

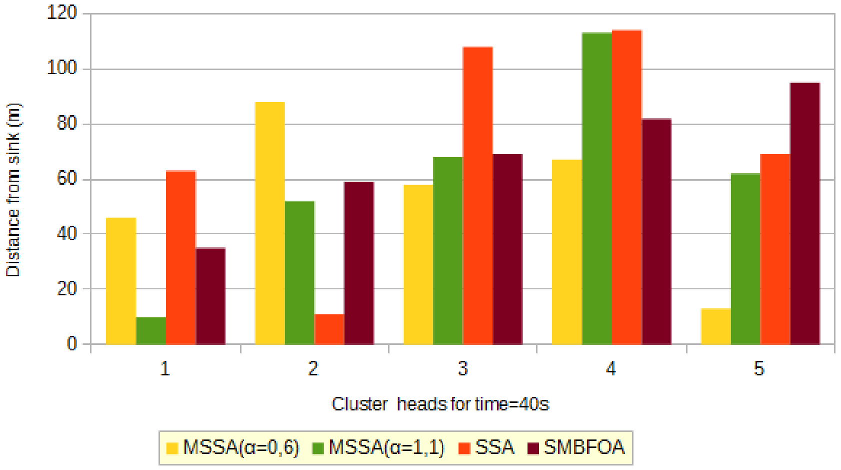


Fig. 10. Distance between sink and CHs at time=20s

Fig. 11 represents the number of alive nodes using the three algorithms SMBFOA, SSA, and MSSA with $\alpha = 0.6$ and $\alpha = 1.1$. It can be noticed from the curves that the lifetime of the network is increased slightly in the case where the MSSA algorithm is used, a difference of 600*s* is added. Also, we can observe that when using the value 0.6 for the α parameter, the lifetime of the network is increased by 100*s* compared to a network that used the value of 1.1 for the α parameter. This can be explained by the fact that the complexity of the MSSA algorithm is lesser than that of the SMBFOA algorithm. Besides, the parameter α adds a certain speed of convergence and efficiency to find the ideal position for the sink.

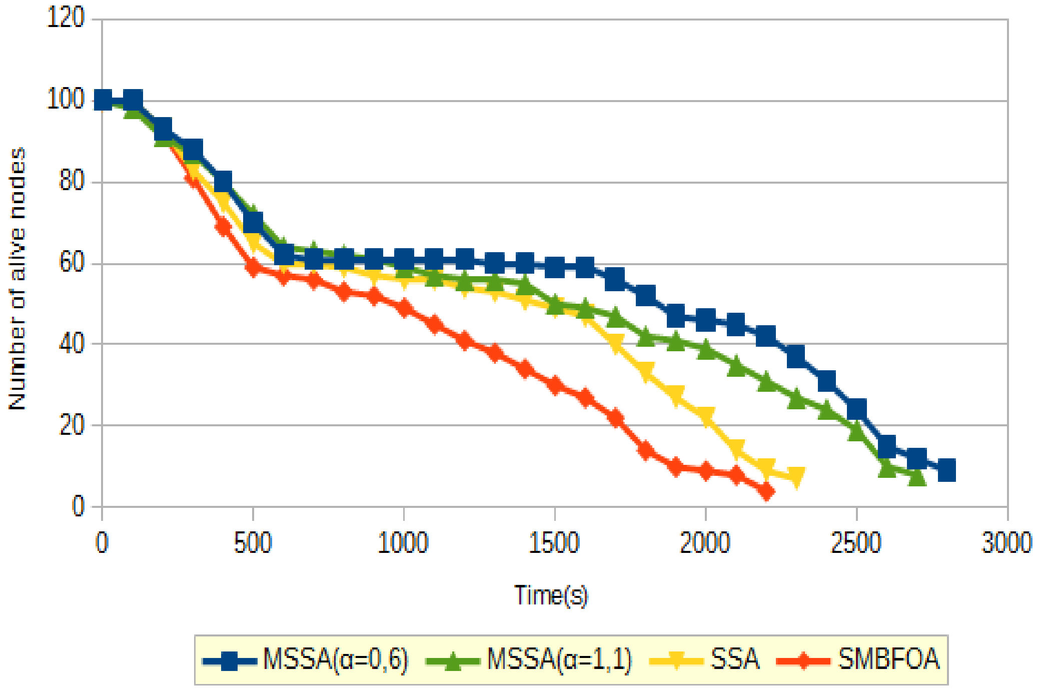


Fig. 11. Number of alive nodes

Fig. 12 represents the amount of data received by the sink using the three algorithms SMBFOA, SSA, and MSSA with $\alpha = 0.6$ and $\alpha = 1.1$. Note that, the amount of data is better, $1/3$ more packets received when using the MSSA algorithm. We also observe that, when using $\alpha = 1.1$, the amount of data received by the sink increases by $1/5$ compared to that observed when $\alpha = 0.6$. This can be justified by the fact that this algorithm employs the parameters c_1 which balances between exploration and exploitation and α which accelerates the convergence speed during the search to find the new sink position.

Fig. 13 represents the energy consumed by the nodes using the three algorithms SMBFOA, SSA, and MSSA with $\alpha = 0.6$ and $\alpha = 1.1$. It can be seen from the curves that, at any instant t, the energy consumed by the network using the MSSA algorithm is always lower than that consumed by the two other algorithms. MSSA finds the best solution in minimal time. Since the position of the sink is optimized, the energy used by the rendezvous points (CHs) during the data transmission is reduced.

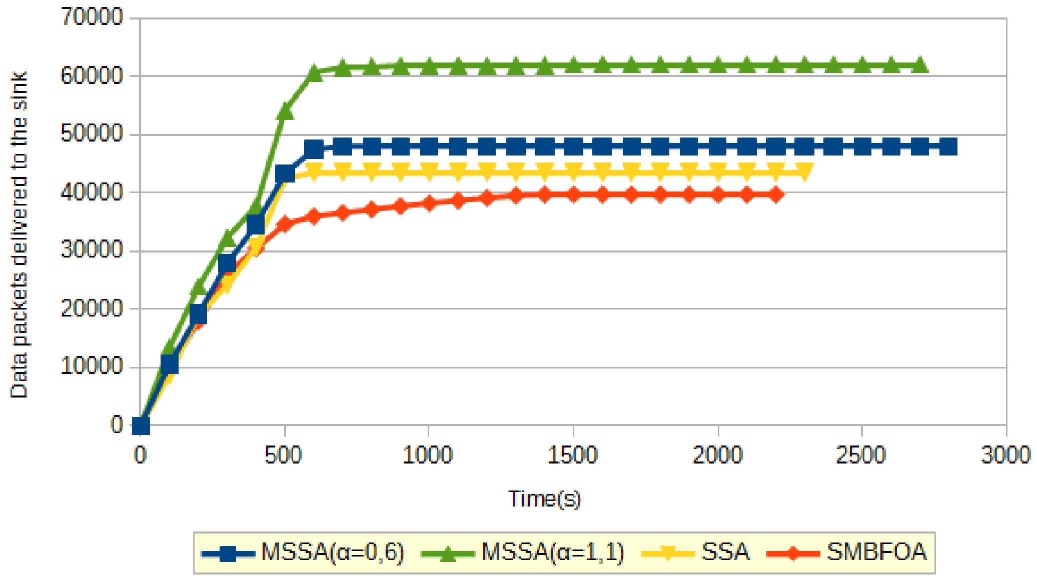


Fig. 12. Data packets received

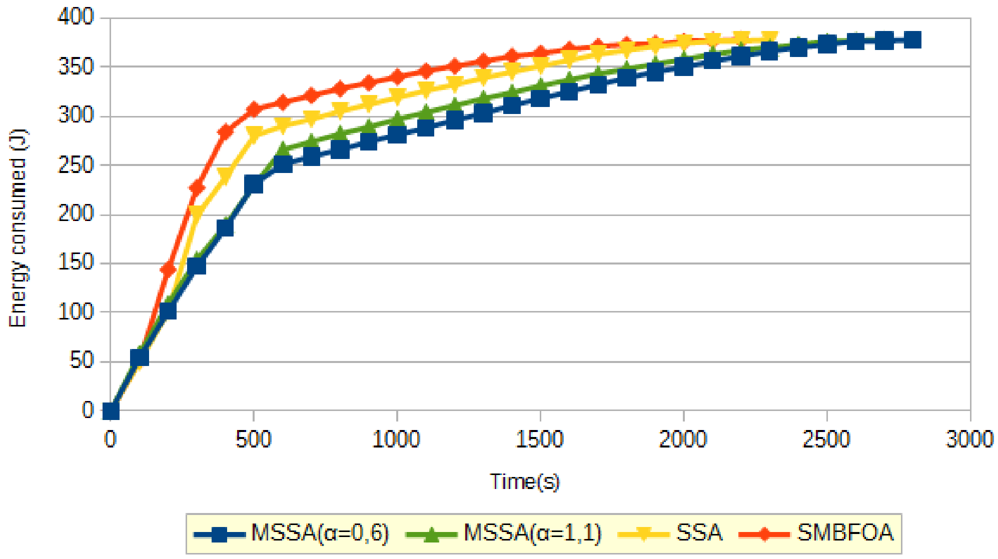


Fig. 13. Energy consumed by nodes

5. Conclusion

In this paper, a modified SSA based mobile data collector for data gathering through heterogeneous IoT network is proposed. The integration of salp's intelligent routing ameliorated the data collection. Simulation results demonstrated that the throughput is extended and lifetime is prolonged when using our proposal. Furthermore, the mobile data collector moved efficiently so that the CH preserves its energy. This results in an improvement of total network lifetime. Moreover, the packet delivery is maximized and thus the buffer overflow problem is minimized. For future work, the proposed approach can be compared with other optimization algorithms.

Acknowledgement

This research work is supported by PHC-Tassili. Grant Number 18MDU114.

References

[1] N. Kumar and D. Dash, “Maximum data gathering through speed control of path- constrained mobile sink in WSN,” in *Proc. of the 7th International Symposium on Embedded Computing and System Design(ISED)*, pp. 1-4, 2017. Article (CrossRef Link)

[2] J. Wang, J. Cao, R. S. Sherratt, and J. H. Park, “An improved ant colony optimization-based approach with mobile sink for wireless sensor networks,” *The Journal of Supercomputing*, pp. 6633-6645, 2017. Article (CrossRef Link)

[3] R. Hamidouche, Z. Aliouat, A. M. Gueroui, A. A. A. Ari, and L. Louail, “Classical and bio-inspired mobility in sensor networks for IoT applications,” *Journal of Network and Computer Applications*, vol. 121, pp. 70-88, 2018. Aritcle (CrossRef Link)

[4] R. Chen, C. Dong, Y. Ye, Z. Chen, and Y. Liu, “QSSA: Quantum evolutionary salp swarm algorithm for mechanical design,” *IEEE Access*, vol. 7, pp. 145 582-145 595, 2019. Article (CrossRef Link)

[5] A. A. Ateya, A. Muthanna, A. Vybornova, A. D. Algarni, A. Abuarqoub, Y. Koucheryavy, and A. Koucheryavy, “Chaotic salp swarm algorithm for SDN multi-controller networks,” *Engineering Science and Technology, an International Journal*, vol. 22, no. 4, pp. 1001-1012, 2019. Article (CrossRef Link)

[6] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, “Salp swarm algorithm: A bio-inspired optimizer for engineering design problems,” *Advances in Engineering Software*, vol. 114, pp. 163-191, 2017. Article (CrossRef Link)

[7] H. M. Kanoosh, E. H. Houssein, and M. M. Selim, “Salp swarm algorithm for node localization in wireless sensor networks,” *Journal of Computer Networks and Communications*, vol. 2019, 2019. Article (CrossRef Link)

[8] Y. Jiang, L. Zhang, and J. Liu, “The path planning of mobile sink based on wolf pack algorithm,” in *Proc. of International Conference on Intelligent Transportation, Big Data & Smart City(ICITBS)*, pp. 147-150, 2019. Article (CrossRef Link)

[9] R. Hamidouche, M. Khentout, Z. Aliouat, A. M. Gueroui, and A. A. A. Ari, “Sink mobility based on bacterial foraging optimization algorithm,” in *Proc. of Computational Intelligence and Its Applications*, vol. 522 , pp. 352-363, 2019. Article (CrossRef Link)

[10] D. Mechta and S. Harous, “Prolonging WSN lifetime using a new scheme for sink moving based on artificial fish swarm algorithm,” in *Proc. of the 2nd International Conference on Advanced Wireless Information, Data, and Communication Technologies*, pp. 1-7, 2017. Article (CrossRef Link)

[11] G. Yogarajan and T. Revathi, “Nature inspired discrete firefly algorithm for optimal mobile data gathering in wireless sensor networks,” *Wireless Networks*, pp. 2993-3007, 2017. Article (CrossRef Link)

[12] M. Krishnan, Y. M. Jung, and S. Yun, “An improved clustering with particle swarm optimization-based mobile sink for wireless sensor networks,” in *Proc. of the 2nd International Conference on Trends in Electronics and Informatics(ICOEI)*, pp. 1024-1028, 2018. Article (CrossRef Link)

[13] Y. Yue, L. Cao, B. Hang, and Z. Luo, “A swarm intelligence algorithm for routing recovery strategy in wireless sensor networks with mobile sink,” *IEEE Access*, vol. 6, pp. 67434-67445, 2018. Article (CrossRef Link)

[14] K. V. K. Stephen and V. Mathivanan, “An energy aware secure wireless network using particle swarm optimization,” in *Proc. of Majan International Conference(MIC)*, pp. 1-6, 2018. Article (CrossRef Link)

[15] G. Ren, J. Wu, and F. Versonnen, “Bee-based reliable data collection for mobile wireless sensor network,” *Cluster Computing*, pp. 9251-9260, 2018. Article (CrossRef Link)

[16] R. Hamidouche, Z. Aliouat, A. A. A. Ari, and M. Gueroui, "An efficient clustering strategy avoiding buffer overflow in IoT sensors: A bio-inspired based approach," *IEEE Access*, vol. 7, pp. 156733-156751, 2019. Article (CrossRef Link)

Ranida Hamidouche received her M.Sc. degree in networks and distributed systems in 2016 from University Ferhat Abbas Setif 1, Algeria. She is currently pursuing the Ph.D. degree with the LRSD Lab of the University of Ferhat Abbes Setif1, Algeria and the LI-PaRAD Lab of the University of Versailles St-Quentin-en-Yvelines, France. Her research interests are focused on bio-inspired computing, Wireless Networks and IoT.

Abdelhak Mourad Gueroui is an associate professor in computer engineering at the University of Versailles, France. His research interests are in the area of wireless networks (WATM, WLAN and sensors), particularly performance evaluation and QoS provisioning. He received his M.Sc. degree from University of Paris6 and his Ph.D. in networking and computer engineering from the University of Versailles.

Zibouda Aliouat obtained her engineering diploma in 1984 and MSc in 1993 from Constantine University. She received her PhD from Setif University of Algeria. She was an Assistant at Constantine University from 1985 to 1994. She is currently an Associate Professor in the Computer Engineering Department at Setif University of Algeria. Her research interests are in the areas of wireless mobile networks modeling and simulation, wireless sensor networks and fault tolerance of embedded systems.

Ado Adamou Abba Ari is a Research Associate at the LI-PaRAD Lab of the University of Versailles St-Quentin-en-Yvelines, France and at the LaRI Lab of the University of Maroua, Cameroon. He received his Ph.D. degree in computer science in 2016 from Université Paris-Saclay in France with the higher honors. He also received the Master degree of Business Administration (MBA) in 2013, the Master of Science (M.Sc.) degree in computer engineering in 2012 and the Bachelor of Science (B.Sc.) degree in mathematics and computer science in 2010 from the University of Ngaoundéré, Cameroon. He served/serving on several journals and conferences program and reviewing committees. Thus, he achieved the outstanding reviewer status with the Elsevier Computer Networks (IF: 2.522). Moreover, he is recognized reviewer of a number of journals including IEEE Access (IF: 4.098), Journal of Network and Computer Applications (IF: 5.273), Computer Communications (IF: 2.766), Remote Sensing (IF: 4.118), Sensors (IF: 3.031), Telecommunication Systems (IF: 1.527), Wireless Personal Communications (IF: 1.200), Sustainable Computing: Informatics and Systems (IF: 1.800), etc. His current research is focused on bio-inspired computing, Wireless Networks, IoT, 5G and the Cloud Radio Access Network.