Research Article

Open Access

CrossMark

Extending the lifetime of wireless sensor networks using a genetic model for defining mobile sinks

Abstract

Mobile Sinks has been widely used in Wireless Sensor Networks (WSNs) to increase the network lifetime, especially to distribute energy consumption among the nodes composing the network. The development of low cost multi–copters, capable of autonomous flights, turns this type of vehicle into a promising solution for acting as mobile sinks. This work presents a new approach to the problem of routing multi–copters to collect data in WSNs in order to extend the network lifetime. We consider that each multi–copter is allowed to stop in a pre–defined number of points (locations) within a monitored area for collecting data from sensors. We propose a genetic model that shows how the distribution of the nodes workload can extend the network lifetime. Simulations are performed to adjust the parameters of the model, and also to evaluate the performance of the proposed strategy.

Keywords: wireless sensor networks, mobile sinks, genetic programming, computer networks, optimization

Volume 2 Issue 6 - 2017

Sergio de Olivera, Pedro M Shiroma, Rone S Ilídio, Marconi A Pereira, Cristiano M Silva

Department of Technology, Federal University of São João Del–Rei, Brazil

Correspondence: Cristiano M Silva, Department of Technology, Federal University of São João Del–Rei, Brazil, Email cristiano@ufsj.edu.br

Received: May 26, 2017 | **Published:** August 17, 2017

Abbreviations: WSN, wireless sensor networks; UAV, unmanned aerial vehicles

Introduction

Wireless Sensor Network (WSNs) are ad hoc networks composed by hundreds (maybe thousands) of sensor nodes spread over a monitored area with the goal of collecting environmental data (such as temperature, light, humidity, and so forth) and forward the collected data to a special node called sink, which is responsible for the interface between user and network. Recently, the literature has presented several research reports on the use of mobile sinks in WSN to increase the network lifetime. Since nodes surrounding the sink tend to consume more energy than peripheral nodes (since they have a large demand for transferring data to the sink), they tend to quickly deplete their batteries. Using mobile sinks, WSN reduce the number of transmissions required from these nodes, consequently increasing the network lifetime.[1–3] Several entities may serve as mobile sinks (humans, animals, land/aerial vehicles). Unmanned Aerial Vehicles (UAV) such as multi–copters can also be used as mobile sinks focusing on increasing the network lifetime by alternating nodes serving as sinks. The ability to hover over predefined locations allows UAV to be efficiently used as mobile sinks.[1] Considering UAV and WSNs with multi–hops data routes delivering data to multi–copters within the monitored area, we propose a novel genetic algorithm to define the sequence of points where the drone has to stop for data collection in order to reduce the network energy consumption while increasing the network lifetime.

Genetic Algorithms are evolutionary approaches designed to solve optimization problems. They start from a set of possible solutions for a problem (initial population) typically created by random or greedy strategies. Then, some operations are applied interactively on this population in order to derive better solutions. After each interaction, the better solutions (or individuals) are chosen to generate another population. The quality of an individual is measured by a fitness function, which is usually calculated by the objective function in the optimization problem. In order to use Genetic Algorithms, we must be able to identify the most important characteristics of the problem, and model them to optimize the algorithm. In the context of this work, the genetic algorithm improves the solution by finding more interesting points for the data collection in a stop–and–go approach. Here, we consider a finite number of sensor nodes, which creates a connected WSN over a limited monitored area. All nodes have constant radio range and they consume energy according to the amount of transmitted packets. The UAV considered is a multi–copter with flying autonomy limited by its batteries. To collect data, the drone has to hover for some period of time in order to wait for the data routes definition, gathering data from the sensor nodes. We consider that the drone must hover over some locations in the monitored area for sufficient time for establishing local routes and to collect all local information at that point. The drone autonomy is given by the flying time (and not by traveled distance). Since, we consider the drone roaming using maximum cruise speed, the drones' autonomy is a function of the number of stop locations, allowing us to model the autonomy in terms of the number of stops.

Our goal is to increase the network lifetime for data collection. The metric is defined here as the number of sensing collecting phases until the first node totally depletes its energy. Each collecting phase can be define as one cycle of the network operation in which all the data hold by sensors are acquired by the drone. Thus, for each cycle of the network, the algorithm must be called to define the stop locations considering the available energy of each sensor. Simulation results show that we can extend the network lifetime in several times when we compare the proposed approach to a baseline based on defining routes according to the physical location of each node. Our main contribution is to present a scalable solution, with results up to 1000 nodes, reaching a promising solution in polynomial time. This solution consider drone characteristics, like flying time and energy, limited energy resources and a continuous flying space, in opposition of others works that use a grid–based space. This text is organized as follows: Section II presents the related work. Section III describes the model and methodology. Section IV details our proposed strategy. Section V presents simulated results and, finally, Section VI presents the final remarks and future works.

Related work

Typically, most of the literatures on mobile sinks consider the communication cost among sensor nodes, assuming that maximizing the network lifetime is similar of minimizing the energy consumption.[4–6] However, this is not the only factor to be analyzed, because we must also balance the residual battery energy of nodes when we intend to maximize the network lifetime. There are some researches that also consider the residual energy in the sensor nodes in order to better distribute the workload among the nodes and, consequently, increase the network lifetime.[7,8] Wang et al.[1] surveyed the main works on mobile sink in WSN. These works are divided in two groups: query–based protocols and location–based protocols. Sensor nodes using query–based protocols wait for queries issued by the mobile sink before sending back the collected data. The query contributes to the definition of data routes. Location–based protocols use nodes' locations to create a virtual infrastructure in which data is forwarded to the sink.

Liang et al.[8] also analyze works dealing with mobile sinks, and authors conclude that:

a. Sink mobility models are simple, since they do not consider the drone autonomy, limited speed and obstacles.

b. Data routing and sink trajectory solutions are independent.

c. Data collection is very fast.

d. Works typically do not analyzed network scalability.

The tendency of many works found in the literature is to treat the mobile sink trajectory problem by defining an Integer Linear Programming model, such as.[2,5,8] However, these models only consider mobile sinks with infinite resources. Luo & Hubaux[7] propose a distribute algorithm that define the sink trajectory through a Poisson distribution. But the trajectories are limited to circles with known radius. Shi et al.[9] present a math model for the trajectory definition of mobile sinks. They propose a reduction to convert this problem from a time–depending problem into a location–depending problem. They also propose to discredit the infinity space of solutions in order to convert it into a finite space. These conversions sustain the premises of our work, which consider the drone autonomy based on the flying time (instead of travel distance). Several heuristics are also presented to define drone tours when the number of nodes is sufficient large (forbidding the search for optimal solutions). Almiani et al.,[5] propose an algorithm to create clusters. Sinds are created accordingly to the location of cluster heads. Such strategy reduces the number of hops in data routes, while extending the network lifetime. However, this algorithm can create some clusters with several nodes, while other clusters may hold only a few ones. Such behavior may unbalance the energy consumption, possibly leading to network disconnections. Liang & Luo[8] propose a heuristic that considers the residual energy of nodes and the communication cost. Initially, the heuristic determines the sequence of points that drones must stop in order to maximize the network lifetime. It uses a maximum number of hops in data routes, which is a parameter that must be configured by the user. Then, the heuristic uses a greed strategy to define the subset of locations that must be further optimized. Romao et al.,[4] consider the node locations to limit the possible points where the drone can stop to collect data. Finally, Silva & Nascimento[10] propose a GRASP–based solution for finding better routes in order to save power and increase network lifetime. They compare their approach with a brute force implementation for small networks (up to 50 nodes), and stationary sink. Differently from the previous approaches, our strategy considers a predefined the number of stop points and network holding up to 1,000 nodes. Table 1 compares our approach to a selected selection of related work.

Table 1 Comparing our proposal to a selection of related works

	Search for a good route	Energy balance	No Discretize or limit solution Set	Up to 1000 nodes	Polinomial heuristic
Our proposal	X	X	X	X	X
MA Ming, et al.[2]			X		
Basagni et al.[3]	X		X		X
Romao et al.[4]	X		X		X
Almiani[5]	X		X		X
Tashtarian et al.[6]	X	X	X		
Luo et al.[7]	X	X			
Liang et al.[8]	X	X			X
Shi[9]	X				
Silva[10]	X	X			X

Formal problem statement

The WSN with mobile sink is defined as the 4–tuple (S,A,r,m), such that:

a. $S = s_1, s_2, \ldots, s_n$ represents the set of n sensors randomly deployed over the monitored area.

b. A represents an L x L monitored area.

c. r represents the radio range of every sensor node (in meters).

$p_i \in A$ represents stop points within the monitored area.

a. m represents the number of stop points for collecting data.

Nodes sense environmental data in cycles (storing data temporarily in memory). For each cycle, the sink has to travel over the monitored and collect data stored in nodes. Each sink travel is a set of points where the sink has to stop. The problem considered here is how to define one set of points for each cycle in order to extend the overall number of cycles. In other words, the problem is to define a set of sink trips $T = t_1, t_2, ..., T_Z$, where $t_i = p_1, p_2, ..., p_m$ is a trip composed by a sequence of points such that every point $p_i \in A$. We assume that the number of points on each trip is predefined, and the drone has sufficient energy to perform the cycle.

b. We define the network lifetime as the time duration from the beginning of the WSN operation until the first node sees their battery totally depleted due to packets transmissions. We also consider that each sensor node has limited energy, which enable them to transmit only a fixed (parameterized) number of packets.

Genetic model

The genetic model uses the concept of population, which is a set of possible solutions to the problem. Each solution is called an individual. The quality of an individual is measured by the fitness function. Some operations are applied on the population in order to improve its quality (i.e., increase the fitness of individuals). Most commons operations are crossover and mutation. Mutation changes the characteristics of one individual per time. The crossover grab characteristics of two individuals, combine them, and then generate two new individuals. Individuals with better fitness have more probability to participate this process. This work defines an individual (gene) as the drone trip, i.e., sequence t_i of stop points. A population T_i is a set of Z drone trips. The fitness function indicates how an individual (route) is better than another. Thus, better fitness indicates low energy consumption and high residual energy after the cycle has been performed. The genetic algorithm proposed here defines a population composed by 20 individuals (or trips) for each nodes collection cycle. Initial individuals are randomly created. Then, mutation is applied over the population to alter one individual. Hence, crossover is also applied to add two new individuals. Then, 20 better individuals are chosen, while the remaining is dropped. On each cycle, the genetic operators are applied *N* times. The best value of *N* is shown in the experiments.

Formally:

Mutation

Consists in moving a stop point p_i (randomly selected) from set t_k, which generates a new individual, t_k. This algorithm randomly chooses an individual from a population. Then, it randomly chooses a point p_i among the points that composes this individual. Finally, it moves this point to position $x_i + \delta_x, y_i + \delta_y$, where x_i, y_i are positions of p_i and δ is a random value such that $0 \geq r$ and r is the radio range. This new individual is placed in the population.

Crossover

Consists in combining two individuals t_i and t_k in order to generate other two individuals t_k and t_l. This operator sorts individuals according to their fitness, and then selects two individuals. However, the chance to select an individual is proportional to its position on the queue. For instance, individuals in position *p* have *p* times more chances to be selected. New individuals have half of the stop points selected from one parent.

Energy exhaustion

After defining the model, the genetic algorithm is executed exhaustively until one of the sensor nodes present energy totally depleted, or until some number of predefined collecting phases (cycles) have been reached. The solution is a set of varied stop points for each collecting phase.

Energy balance

In order to increase the network lifetime, it is fundamental to forward data in routes composed by nodes with more residual energy in order to spend minimum energy in collecting phases. This feature implies in how the sensor nodes route messages to the sink, and how to calculate the fitness evaluation. We consider two parameters:

a. Energy spent in collecting phase is inversely proportional to the fitness evaluation because we need to save power. Furthermore,

b. The residual energy of nodes is directly proportional to the fitness.

The relation between energy spent and residual energy in a collecting phase must be such that the residual energy must be as balanced as possible after each cycle. Paths with more residual energy are chosen, which creates balanced solutions. The relation between residual energy in a path and the energy needed to forward a packet in this path is the major issue for finding balanced solutions. We include parameter á, and use the relation $R_i = \sum_{j=0}^{h} \frac{1}{E_j^{á}}$, where h is the path from node to sink and E_j is the residual energy of node j. In subsection 4, we show that $\alpha = 3$ seems to be a good value for our heuristic.

Figure 1 presents an example of how a routing path is chosen. Node S is the sink, while node X has data to send to the sink. Remaining nodes in Figure 1 are labeled with its residual energy. Node X has three possible routes to send data. The first route through node 5, has only a node with residual energy 5. Hence the residual energy of the path is $\frac{1}{5^3} = 0.008$. The second route (via node) 3 has residual energy equal to $\frac{1}{3^3} = 0.037$, while the third route (via nodes 8 and 9 has residual energy $\frac{1}{8^3} + \frac{1}{9^3} = 0.003$.8

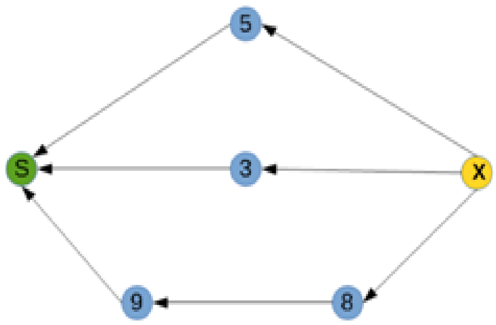


Figure 1 Energy balance in routing example.

We use the same scheme to calculate the fitness. The fitness evaluation for a solution is calculated by summing the cost of every transmission performed by the network. The relative cost of each transmission is the inverse of the residual energy in the transmitter raised by á. The fitness F_i of an individual t_i is calculated by the following equation:

Citation: Olivera SD, Shiroma PM, Ilídio RS, et al. Extending the lifetime of wireless sensor networks using a genetic model for defining mobile sinks. *Int Rob Auto J.* 2017;2(6):226‒232. DOI: 10.15406/iratj.2017.02.00040

$$F_i = \sum_{j=0}^{n} \frac{e_j}{E_{ji}^{\alpha}}$$

Where E_{ji} is the residual energy of node s_j after the drone has performed trip t_i while e_j is the energy spent by node s_j during trip F_i. á is the factor of residual energy appreciation, which defines how much the residual energy will influence in the fitness calculation. Usually, nearby nodes have similar energy levels, since they have similar workload. Moreover, nodes tend to choose the shortest data route. However, with the decrease of the energy level, drones can have more stop points nearby borders of the monitored area. Such locations tend to consume more energy since are composed by more nodes than central routes.

Implementation

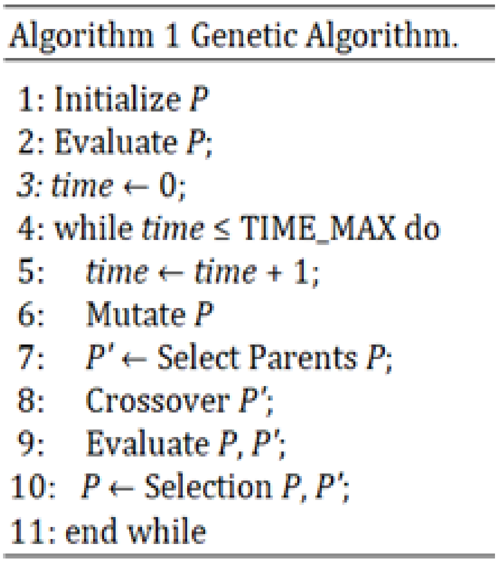
The genetic model is implemented in the Anaconda environment in Python,[11] and it was used to evaluate the stop points of mobile sink. The test mass was generated in a random way, accordingly to the literature. It is composed by a set of Cartesian coordinates representing locations and residual energy of each node. The genetic model, presented in Algorithm 1 includes mutation, crossover and fitness, as presented in the model definition. Also, we implement initialization and selection function to complete the genetic algorithm. According to Algorithm 1, the mutation includes a random choice of a point do be translated for other position of the monitored area by δ_{xy} randomly selected between $(-r,+r)$ meaning an offset from the original position to another one, enough to create a different solution. After mutation, the new individual is included in P, increasing the set with one individual. Complementarily, crossover starts choosing two possible solutions to be combined. After the crossover, the new two children are added in the set P'. Furthermore, the function Evaluate calculates the fitness for all individuals in the network using Equation 1. After that, function Selection selects the better evaluated individuals from subsets P and P' to generate a new set P to restart the process until time generations are met. The algorithm runs considering a test mass, and we are concerned on maximizing the network lifetime (Figure 2). The test mass varies from 50 up to 1,000 nodes. The test mass is generated before the simulation and used for all simulations. The evaluate function test the solution P to find the residual energy of the drone trip. Solutions with more residual energy have higher score, according to Genetic Model section.

Results and discussion

The first simulation was performed with the objective of identifying the best value for α factor in the fitness function. Such function has to maximize the network lifetime by reducing and balancing the energy consumption. We assume that each transmission consumes the same amount of energy. In the simulations, we vary the value of α from 1 to 5. We consider 50 nodes randomly deployed in the monitored area, each one having 50 units of energy able to send 50 data packages. We consider 20 generations on each collecting cycle. Figure 3 presents the results in a plot, while Table 2 highlights the numeric values. We verify function $1/x^3$ presenting the best results for the test mass considered, with small variations for the functions $1/x^2$, $1/x^4$ and $1/x^5$.

Table 2 Fitness selection

Fitness function	Average	Standard deviation
$1/x$	29.3	0.92
$1/x^2$	32.87	0.63
$1/x^3$	33.8	0.61
$1/x^4$	33.33	0.88
$1/x^5$	33.4	0.86



Algorithm 1 Genetic Algorithm.

```
Initialize $P$
Evaluate $P$;
$time \leftarrow 0$;
while $time \leq$ TIME_MAX do
    $time \leftarrow time + 1$;
    Mutate $P$
    $P' \leftarrow$ Select Parents $P$;
    Crossover $P'$;
    Evaluate $P, P'$;
    $P \leftarrow$ Selection $P, P'$;
end while
```

Figure 2 The algorithm runs considering a test mass.

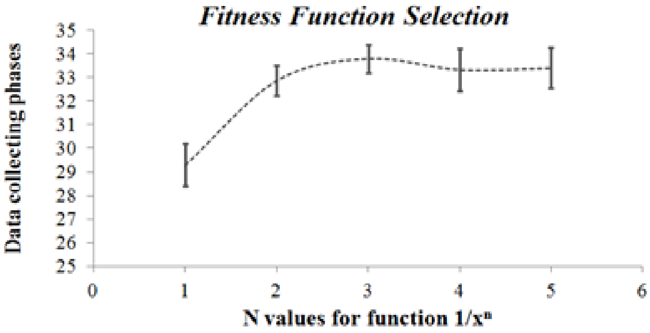


Figure 3 Fitness selection.

Experiment #2: energy consumption

We performed simulations to find how much energy is spent to collect one sensing data from each node in the network in a single cycle. To evaluate our approach, we use the K–Means clustering algorithm,[12] which finds k stop points by clustering the sensor nodes in k groups. The K–Means approach is similar to our implementation because it returns k stop points, so we use it as our baseline. Figure 4 shows results from our simulations. The scenarios simulated have the following parameters:

a. Number of nodes: 100–1000
b. Monitored area: 707x707–2236x2236 (fixed density of 1 node/5.000 measure square units)
c. Initial energy of the nodes: 100 units, since the transmission cost is fixed in one unit.
d. Number of stop points for the sink: 4
e. Reach of mobile sink radio: 200
f. Comparative parameter: Network size, in number of nodes and monitored area
g. Stop condition: Battery exhaustion of one node

We simulate four ways to distribute k stop points within the monitored area:

a. Grid: The points are distributed in a uniform way, spaciously along the monitored area;
b. Centroid: The points are fixed in point positions by the K–means clustering algorithm
c. Genetic 20G: The points are fixed in positions given by the Genetic Algorithm in 20 generations
d. Genetic 40G: The points are fixed in positions given by the Genetic Algorithm in 40 generations
e. Genetic 100G: The points are fixed in positions given by the Genetic Algorithm in 100 generations

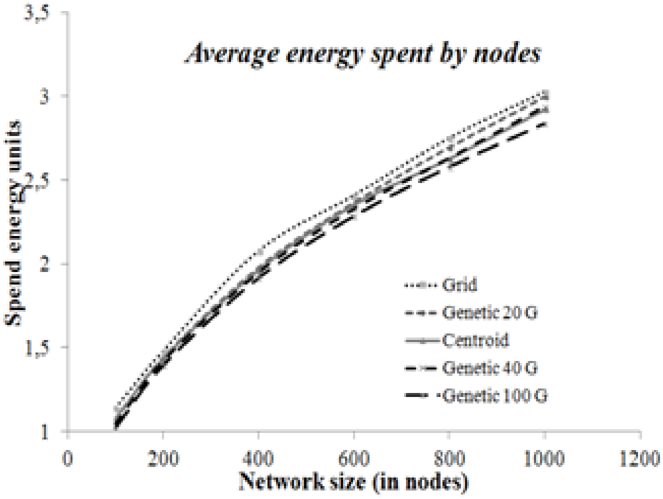


Figure 4 Results of average energy spent in a single sensing phase.

Table 3 shows the obtained results expressed in energy units versus per number of nodes. The results show Genetic 20 G a little worse than K–means points, with Genetic 40G close to K–means and Genetic 100 G better than k–means. The Euclidean grid distribution is the worst because is the unique that doesn't consider locations of nodes. Simulations with 100 nodes spent about one energy unit per node because almost all nodes sent data directly to the mobile sync in a single–hop approach. With few nodes the results tend to be the same since the covered area per mobile sinks is very close to the monitored area. When considering 100 nodes, the covered area is $4x200^2 x\pi = 502,400$ and so almost all nodes send data in a single–hop basis.

Experiment #3: network lifetime

The main objective of this work is to extend the lifetime of the network, given by the time until the first node die. Our work was projected to balance the network in order to increase the Lifetime. The advantage of genetic approach is the possibility of include balance conditions in fitness calculation. Thus, we present how balanced energy fitness can increase the network lifetime. These simulations have the goal of showing the relation between battery capacity and network lifetime. With few nodes, in a little area, low battery capacity is enough to maintain the network for an interesting time interval. However, in large areas having lots of sensor nodes, we demand higher capacity. So, here we simulate the network to find the amount of battery capacity needed to increase the network lifetime in terms of cycles. The genetic model uses 40 generations.

The parameters considered in this simulation are:

a. Number of nodes: 100–500
b. Monitored area: 707x707–1581x1581 (fixed density of 1 node/5,000 measure square units
c. Initial energy of the nodes: 100 units
d. Number of stopping points for the sink: 4
e. Reach of mobile sink radio: 200
f. Comparative parameter: Network size, in number of nodes and monitored area
g. Stop condition: 6 (six) sensing phases

We simulate all network scenarios during six cycles (i.e., the algorithms runs six times). We select this number because it is enough to indicate the differences between the evaluated strategies. Figure 5 shows the minimum initial battery energy to maintain all nodes working up to six sensing phases. It shows that defining the stop points using K–Means requires more energy that our approach, because the genetic algorithm uses a balanced fitness function, while K–Means does not consider battery charges to select the stop points. Numeric results are in Table 4.

Experiment #4: Improving the results through more generations

In the last simulation, we started the tests to show how the genetic model improves the results with more generations. The main expected result for these tests was the number collecting cycles, which is directly related with the network lifetime. As large is the network lifetime, more collecting cycles will be performed. Hence, a sequence of tests was performed in order to show the behavior of this metric in different network topology.

The testes consider the following configurations:

h. Number of nodes: 50
i. Monitored area: 1000 x 1000

j. Initial energy of the nodes: 100 units, since the transmission cost is fixed in one unit.

k. Number of stopping points for the sink: 4

l. Reach of mobile sink radio: 200 units

m. Comparative parameter: number of generations of the genetic algorithm.

Table 3 Average energy spent by nodes in a single sensing phase

# Nodes	Grid	Centroid	Std dev	Genetic 20 G	Std dev	Genetic 40 G	Std dev	Genetic 100 G	Std dev
100	1.14	1.08	0.08	1.07	0.02	1.05	0.02	1.02	0.01
200	1.48	1.43	0.03	1.44	0.02	1.42	0.02	1.4	0.01
400	2.08	1.96	0.02	1.98	0.02	1.95	0.01	1.92	0.01
600	2.41	2.35	0.01	2.36	0.02	2.33	0.02	2.28	0.01
800	2.75	2.63	0	2.69	0.02	2.63	0.02	2.58	0.01
1000	3.03	2.92	0.01	2.99	0.02	2.94	0.26	2.83	0.02

Table 4 Results for six cycles

Number of nodes	Bat. charge	Std dev	Bat. charge	Std dev
100	12	1.08	3.13	0.43
200	46.7	9.6	12.5	1.81
300	92.2	10.17	18.63	2.41
400	108.63	12.49	26.23	3.22
500	152.73	23.02	34.67	3.92

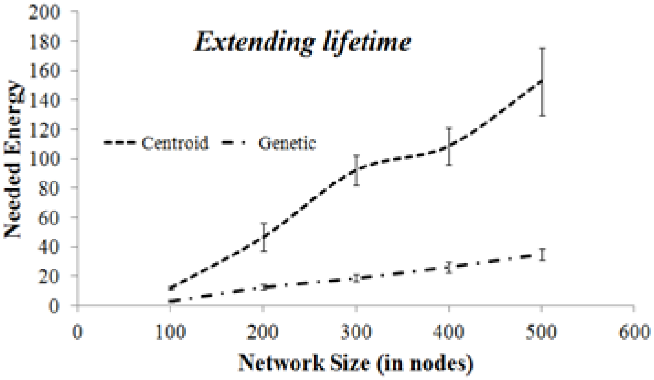


Figure 5 Results of energy needed in six sensing phases.

The baseline considered here is when no genetic operation is performed and the sink positioning is totally random. Figure 6 shows the results comparing the network lifetime recorded by the number of cycles (as a function of the number of generations ran by the genetic algorithm). It can be seen an increase of the network lifetime from the first measures to the last ones. Comparing this with the random solution recorded as "zero" genetic generations in Table 5, the network lifetime increases up to 200%. It can be observed a network lifetime close to 75 collecting phases indicating a possible limit to an optimum solution. Considering the resources of 100 transmission packages per node, and cost of forwarding messages to the sink, this solution may be considered a promising one. The theoretical maximum of 100 collecting phases only can be reached if all nodes send data directly to a sink position, what is not possible to the test mass used. This possibility does not represent an issue at all, since it can easily solved sending all collected data directly to the sinks.

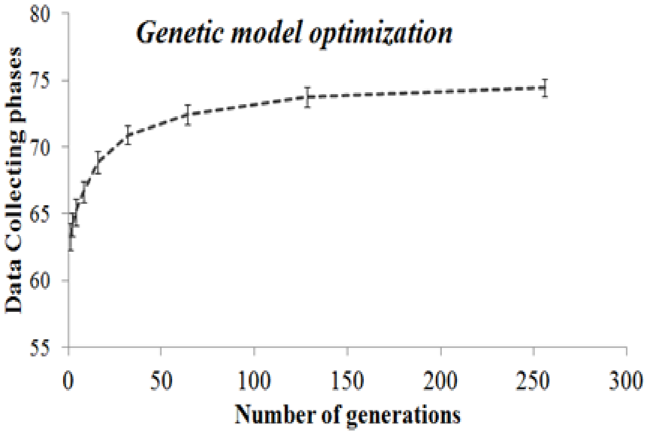


Figure 6 Results of genetic model optimization.

Table 5 Results of genetic model optimization

Generations	Average	Standard deviation
0	27,53	3,95
1	63,27	1,01
2	64,17	0,87
4	65,13	1,01
8	66,63	0,81
16	68,83	0,83
32	70,90	0,71
64	72,43	0,73
128	73,77	0,73
256	74,43	0,63

Conclusion

This work presented the implementation of a genetic model to find the stop points for mobile sinks in wireless sensor networks with the goal of balancing energy consumption for increasing the network lifetime. Our approach is differently from the ones reported by the literature by proposing a balanced selection of stop points while increasing the network lifetime. We stress the network until the end of the lifetime and we show how a balanced approach may increase the lifetime. Our work considers multi–copters as mobile sinks capable to stop on some predefined locations in order to receive data from surrounding sensors. The displacement between two adjacent stop points is not considered since the drones autonomy is based on the flying time, and not on travel distance. Hence, the drone limitation is the number of stopping points. The results show the genetic model as a promising alternative for defining stop points to minimize the network energy consumption. The fitness function is defined to balance the network energy, which is required for increasing the network lifetime by consuming energy in a uniform way. As future work, we intend to adapt this approach for using the Traveling Salesman Problem for defining the stop points.

Acknowledgments

None.

Conflict of interest

Author declares that there is none of the conflicts.

References

1. WANG J, Yang X, Zhang Z, et al. A Survey About Routing Protocols with Mobile Sink for Wireless Sensor Network. International Journal of Future. *Generation Communication and Networking*. 2014;7(5):221–228.
2. MA Ming, Yuanyuan Yang, Miao Zhao. Tour Planning for Mobile Data–Gathering Mechanisms in Wireless Sensor Networks. *IEEE T Vehicular Technology*. 2013;62(4):1472–1483.
3. Basagni Stefano, Alessio Carosi, Emanuel Melachrinoudis, et al. Controlled Sink Mobility for Prolonging Wireless Sensor Networks Lifetime. *Wireless Networks*. 2008;14(6):831–858.
4. Romao, Oberlan C, Andre G dos Santos, Geraldo R Mateus. Lifetime Maximization of Hop–and–Delay Constrained Wireless Sensor Networks with Mobile Agent. *IEEE Congress on Evolutionary Computation*. 2013.
5. Almiani K, A Viglas, L Libman. Energy–Efficient Data Gathering with Tour Length–Constrained Mobile Elements in Wireless Sensor Networks. *39th Annual IEEE Conference on Local Computer Networks*. 2010. p. 1–8.
6. Tashtarian F, Moghaddam MHY, Effati S. *Energy Efficient Data Gathering Algorithm in Hierarchical Wireless Sensor Networks with Mobile Sinks*. International Econference on Computer and Knowledge Engineering. 2012.
7. Luo Jun, Jean Pierre Hubaux. Joint Sink Mobility and Routing to Maximize the Lifetime of Wireless Sensor Networks: The Case of Constrained Mobility. *IEEE/ACM Trans Netw*. 2010;18(3):871–884.
8. Liang W, LUO J. Network lifetime maximization in sensor networks with multiple mobile sinks. *LCN IEEE Computer Society*. 2011. p. 350–357.
9. Shi YTY, Hou. Some Fundamental Results on Base Station Movement Problem for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*. 2012;20(4):1054–1067.
10. Silva, Rone Ilidio da, Mario A Nascimento. On Best Drone Tour Plans for Data Collection in Wireless Sensor Network. *Proceedings of the 31st ACM Symposium on Applied Computing*. 2016. p. 703–308.
11. Martins L Felipe. I Python Notebook Essentials. *Packt Publishing–ebooks Account*. 2014.
12. Lloyd, Stuart P. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*. 1982;28(2):129–137.