

A Genetic Algorithm Based and Subtree Restricted Routing Algorithm in Wireless Sensor Networks

<https://doi.org/10.3991/ijoe.v14i03.8018>

Zhi Huang

Mianyang Teachers' College, Mianyang, Sichuan, China

huangzhi2003@126.com

Abstract—The GA (genetic algorithms) was applied to find routing trees with minimal maximal sensor load thus prolong network lifetime in wireless sensor networks. However, the existing algorithm limits search space of GA to avoid generating cycles by dividing sensors into different layers and stipulating that a sensor can only select a sensor in its lower layer as its new parent. In this paper, we have found that how cycles are generated during GA operations, and proposed a solution based on subtree to avoid generating cycles. The solution only requires a sensor not to change its parent to a sensor in the subtree with it as the root. And layers are no longer considered. As a result, GA has more search space and GA operations can be performed more freely. Thus more optimal routing trees can be computed and network lifetime can be extended. The experiment shows that the proposed algorithm prolongs network lifetime notably.

Keywords—Wireless sensor networks, Network lifetime, Routing tree, Genetic algorithm, Maximal sensor load, Subtree

1 Introduction

Wireless sensor networks have gradually become a key technique that is widely used in military, environmental protection, industry, agriculture, space exploration, and transportation, etc. [1-9]. Energy of sensors is usually provided by lightweight batteries. Once one or a proportion of sensors deplete their energy, the whole network loses its efficacy. Therefore, limited energy is a bottle-neck of wireless sensor networks. And how to use sensors' energy intelligently and sufficiently thereby extend network lifetime is an important issue in the research of wireless sensor networks [8-17].

Many researchers have studied routing algorithms to polish up energy efficiency. And many routing algorithms have been researched to prolong network lifetime by improving energy efficiency. The proposed routing algorithms are of two sorts: static ones and dynamic ones. Using static routing algorithms, only one routing is established and employed during the entire network duration. Using dynamic routing algorithms, multiple routings are dynamically constructed and employed based on current network status. Dynamic routing algorithms can change according to constantly

changing status of a network, but they need to obtain information of current network status and are only suitable for the applications meeting some specific conditions. For instance, some dynamic routing algorithms require all sensors periodically inform their current residual energy to the sink. And they are only suitable for applications in which the time between two times of data collecting is sufficient so that sensors have enough time to inform the sink their current residual energy. However, static routing algorithms can be adopted by all applications owing to that it is unnecessary to obtain information of changing network status and applications aren't required to meet specific conditions [9, 18, 19].

In some applications, that even only one sensor exhausts its energy results in the failure of the network. Therefore, their network lifetimes are defined as the time when the first sensor exhausts its energy. In other applications, the network loses its efficacy only when a proportion of sensors exhaust their energy. And their network lifetime is defined as the time when a certain proportion of sensors exhaust their energy [6].

Usually, a network is considered as a graph. And the weight of the link between two sensors is the energy expended to transfer a data packet between them [18, 19]. Proposed static routing algorithms aim at constructing an optimal routing tree to improve energy efficiency. With Power Efficient Data Gathering and Aggregation Protocol (PEDAP), a routing tree is constructed based on Minimum Spanning Tree (MST) [18]. Another routing algorithm, Least Energy Tree (LET) was proposed in [14]. LET regards the sink as the root and establishes a minimum path tree. Therefore, the energy expended each sensor forwards its data packets to the sink is minimum and the network's total energy consumption is also minimum [19].

In [9], it is pointed out that a sensor depletes the same energy in every data gathering because of unchangeable routings with static routing algorithms. And a sensor's energy consumption in every data gathering is defined as its load. Furthermore, it is pointed out that in a network with the time the first sensor exhausts its energy as its network lifetime, what decides network lifetime is not the total energy consumption of the network, but the maximal sensor load in the network. And a GA (Genetic Algorithm) based routing algorithm aiming at minimizing the maximal sensor load, named as GA Based Minimal Maximal Load Tree (GA-MMLT), was proposed. In GA-MMLT, a routing tree is encoded as a chromosome with each sensor's parent sensor as a gene. To avoid producing cycles during GA operations, all sensors are partitioned into layers in accordance with their hops from the sink. And GA operations are only implemented between adjacent layers, i.e. a sensor only can select its new parent in its lower layer [9]. However, this limits search space of GA and the performance of the algorithm.

We have proposed an algorithm based on GA-MMLT in this paper. We have studied how cycles are generated and found that cycles aren't generated during GA operations only if a sensor doesn't change its parent to a sensor in its current subtree. Therefore, the proposed algorithm is named as SR-GA-MMLT (Subtree Restricted GA-MMLT). With SR-GA-MMLT, it's non-essential to divide sensors into layers. During GA operations, two rules are observed: 1) when crossover is operated on a sensor, the algorithm firstly checks whether the sensor's new parent is in its current subtree. If the new parent is in its current subtree, the crossover is not permitted; 2) in mutation

operation, a sensor selects its new parent from the sensors which is its neighbors and not in its current subtree. The rules above ensure that any cycles won't be generated during GA operations. Therefore, GA operations can be performed more freely and has more search space compared to GA-MMLT. As a result, network lifetime is notably extended.

The structure of the remainder of the paper is as follows: Related studies of routing algorithms are described in Section 2; in Section 3, the model of our work is presented; in Section 4, we describe the proposed algorithm in detail; in Section 5, we provide and analyze experimental results; finally, in Section 6, we have concluded this paper.

2 Related Works

We summarize some typical routing algorithms in this section.

In [18], Power Efficient Data Gathering and Aggregation Protocol (PEDAP) was proposed. In PEDAP, the energy expended to transfer a data packet between two sensors is set as the weight of the link between them. And a routing tree is constructed based on Minimum Spanning Tree (MST) algorithm. However, routing trees constructed with this routing algorithm are not energy efficient because it doesn't consider the destination of data packets [18].

Power Efficient Data Gathering and Aggregation Protocol-Power Aware (PEDAP-PA) was proposed based on PEDAP. Using PEDAP-PA, the sink estimates residual energy of each sensor and updates the weights of links with the following equation periodically [18]:

$$w(i, j) = \quad (1)$$

where $c(i, j)$ is the energy expended to send a data packet from sensor i to sensor j . And $e(i)$ is the estimated residual energy of sensor i [18]. The algorithm considers residual energy of sensors. However, this routing algorithm is dynamic and only suitable for the applications meeting specific conditions.

In [19], it is illuminated that the routing trees based on MST (Minimum Spanning Tree) algorithm are not optimal. The optimal routing trees should be the ones constructed with Least Energy Tree (LET) which treats the sink as the root and constructs a minimum path tree. The algorithm resumes that data gathering is not frequent and the time between two times of data gathering is enough. Therefore, sensors have enough time to inform the sink their residual energy. Then the sink updates the weights of links with the following equations [19]:

$$w(i, j) = [c(i, j)]^\alpha \left[\frac{1}{e(i)} \right]^\beta \quad (2)$$

$$w(i, j) = \gamma c(i, j) + \delta \frac{1}{e(i)} \quad (3)$$

where $\alpha, \beta, \gamma, \delta$ are positive constant and $\gamma + \delta = 1$. The former is named as ratio-w and the latter is named as sum-w. And then a new routing tree is constructed and

employed [19]. The algorithm is also a dynamic routing algorithm and requires applications to meet specific conditions. Therefore, it is not suitable for all applications.

In [11], an algorithm based on GA was proposed for two-tiered networks with the sensors as the bottom tier and the relay nodes having higher energy as the upper tier. Some sensors and one relay node form a cluster in which the relay node acts as cluster head, receives data packets from its sensors and sends them to the sink with the network constituted by all relay nodes. Network lifetime is determined by relay nodes. Relay nodes' data gathering is scheduled by the proposed algorithm based on GA to extend network lifetime [11]. However, the algorithm is proposed for networks with specific two-tiered structure.

GA is also employed to choose cluster headers in [20]. The whole network area is partitioned into some areas involving some sensors. In every area, some sensors serve as cluster headers. GA is employed to find appropriate solutions of cluster headers in every area for each round to extend network lifetime [20]. However, the algorithm is designed for clustering structure networks.

In [9], it is pointed out that in a network with the time the first sensor exhausts its energy as its network lifetime, what determines network lifetime is the maximal sensor load in the network. And a GA based routing algorithm named as GA-MMLT was proposed to minimize the maximal sensor load, in which a routing tree is encoded as a chromosome with each sensor's parent sensor as a gene [9]. However, all sensors are divided into layers according to their hops from the sink. And GA operations are restricted to be performed between adjacent layers to avoid producing cycles [9]. Therefore, search space of GA and the performance of the algorithm are limited.

3 The Model

A square network area with some homogenous sensors and one stationary sink is considered in our work. The lightweight batteries powered sensors are randomly and evenly deployed [9]. And the sink which is supposed to have infinite memory, computing power and energy is deployed at the center of the network [9, 21]. Once the network is deployed, each sensor obtains its position with some technologies, such as GPS, and reports its position to the sink [21].

We have employed the first order radio model [22] as our radio model. In the model, the energy expended to receive a data packet with k bits is computed with the following equation [19, 22]:

$$C_R(k) = kE_{elec} \quad (4)$$

where E_{elec} is the energy expended by transmitter to send one bit or receiver circuits to receive one bit. And the energy expended to send a data packet with k bits is computed with the following equation [19, 22]:

$$C_T(k, d) = kE_{elec} + kE_{elec}d^\gamma \quad (5)$$

where d is the distance between the two sensors. And $E_{elec}d^\gamma$ is the energy expended by transmit amplifier sending one bit in which γ is path loss exponent which is

usually 2, 3, or 4. So the energy expended to transfer a data packet with k bits between two sensors is computed with the following equation [19, 22]:

$$C(k, d) = kE_{elec} + kE_{elec}d^\gamma \tag{6}$$

4 The Genetic Algorithm Based and Subtree Restricted Routing Algorithm

As mentioned before, it is pointed out that with static routing algorithm, a sensor consumes the same energy because the routing is static [9]. Therefore, a sensor's energy consumption in every data gathering can be defined as its load. And in a network with the time the first sensor exhausts its energy as its network lifetime, what determines network lifetime is not the total energy consumption of the entire network, but the maximal sensor load of the network. Network lifetime can be computed as the following equation [9]:

$$L = \frac{e}{L_{max}} \tag{7}$$

where e is initial energy of sensors. And L_{max} is the maximal sensor load in the network. GA-MMLT employs GA to reduce L_{max} thereby extend network lifetime. In detail, a routing tree is encoded as a chromosome with each sensor's parent sensor as a gene, as shown in Fig. 1 [9].

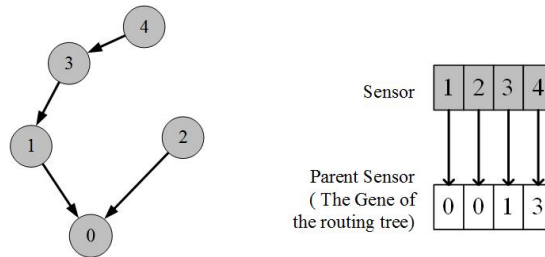


Fig. 1. A routing tree is encoded as a chromosome[9]

Network usually involves many sensors and has very complicated topology. Therefore, cycles may be generated during crossover or mutation. Even if only one cycle is generated, the network becomes disconnected. This result in some sensors can't forward their data packets to the sink. To avoid generating cycles, GA-MMLT divides all sensors into different layers according to their hops from the sink, and stipulates that crossover or mutation is only implemented between adjacent layers. This ensures that any cycles aren't generated.

Dividing sensors into different layers and stipulating that GA operations are only performed between adjacent layers limits search space of GA and obtaining more optimal routing trees. As shown in Fig. 2, (a) is the network topology in which the sink is node 0, and other nodes are sensors. The number beside a link is the energy

expanded to send a data packet on it, i.e. its weight. And the number in the brackets beside a node is the layer of the sensor, i.e. its hops from the sink. (b) is the routing constructed with GA-MMLT. Since sensor 4 is in layer 2, it only can select the sensors in layer 1, such as sensor 2, as its parent. And the maximal sensor load is $4 \times (1+1) = 8$ at sensor 2. However, if sensor 4 selects sensor 3 as its parent, the maximal load is $2 \times (1+1+1) = 6$ at sensor 1, as shown in (c).

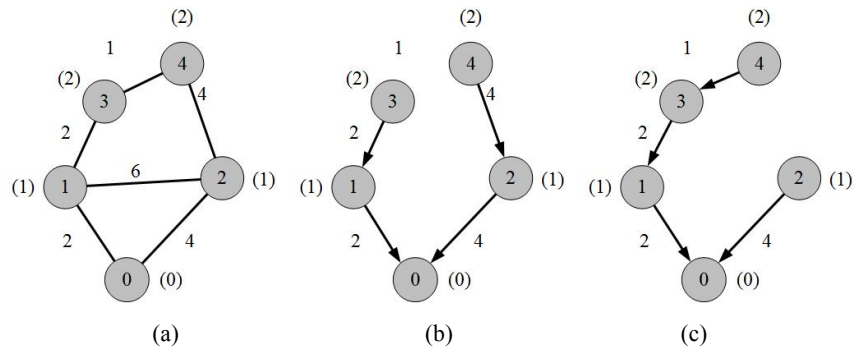
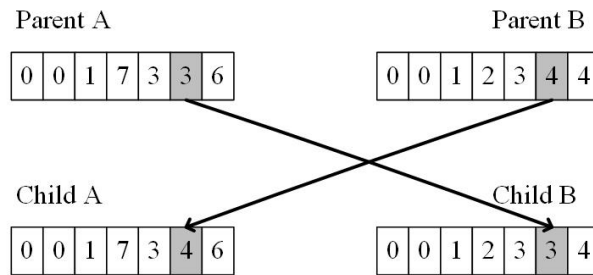


Fig. 2. The existing algorithm limits search space of GA and generating more optimal routing trees

However, if we don't divide sensors into different layers and stipulate that GA operations are only performed between adjacent layers, cycles may be generated during crossover or mutation. As shown in Fig. 3, two routings, A and B are being crossover operated, where the 6th gene is chosen to crossover. And this leads to generating a cycle. Cycles may also be generated during mutation, as shown in Fig. 4. In Fig. 4, the 4th gene is chosen to perform mutation. After the operation, a cycle is generated. As a result of generating cycles, the network becomes disconnected and some sensors can't send their data packets to the sink.



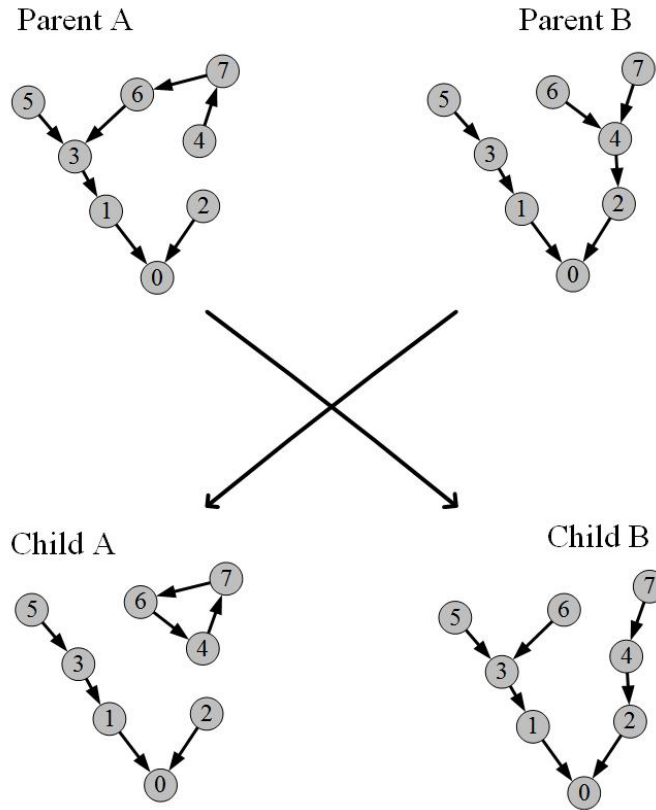


Fig. 3. A cycle is generated during crossover operation [9]

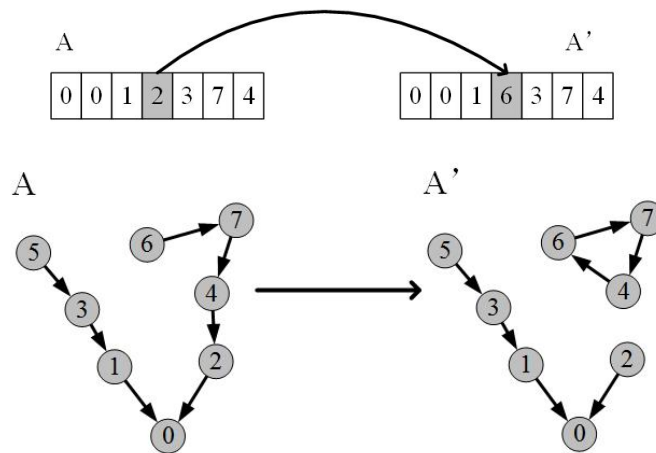


Fig. 4. A cycle is generated during mutation operation [9]

We have found that cycles are generated during GA operation because a sensor selects a sensor in the subtree with it as the root as its new parent. For example, in Fig. 4, in the routing tree of Parent A, sensor 4 is in the subtree with sensor 6 as the root. After sensor 6 selects sensor 4 as its new parent, a cycle is generated. Similarly, in Fig. 4, in the routing tree of Parent A', sensor 6 is in the subtree with sensor 4 as its root. After sensor 4 selects sensor 6 as its new parent, a cycle is generated.

Therefore, if only a sensor doesn't change its parent to a sensor in the subtree with it as the root, it is impossible to generate a cycle. On the contrary, if a sensor changes its parent to a sensor in the subtree with it as the root, a cycle must be generated. Considering that a connected routing tree without cycles, in which the sink is s , sensor k changes its parent to sensor i , which is not in the subtree with k as the root. Since there must be a path $i \rightarrow i_1 \rightarrow i_2 \rightarrow \dots \rightarrow s$, and k is not in the path, it is certain that there is a path $k \rightarrow i \rightarrow i_1 \rightarrow i_2 \rightarrow \dots \rightarrow s$. Therefore, no cycle is generated and the routing tree is still connected. On the contrary, if k changes its parent to sensor j , which is in the subtree with k as the root, there must be a path $k \rightarrow j \rightarrow j_1 \rightarrow j_2 \rightarrow \dots \rightarrow k$. Therefore, a cycle is generated and the routing becomes unconnected.

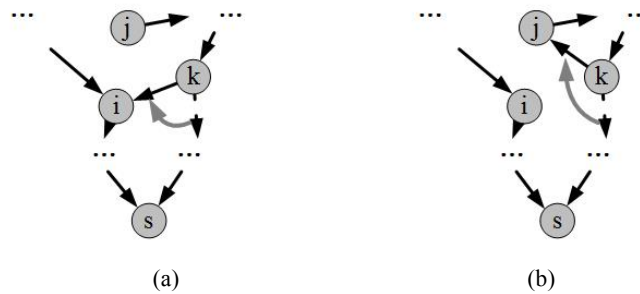


Fig. 5. How no cycle is generated and how a cycle is generated

To avoid generating cycles, the following two rules during GA operations are constituted:

1. In crossover operation on a sensor, whether the sensor's new parent is in its current subtree is firstly checked. If the new parent is in its current subtree, the crossover operation isn't permitted. On the contrary, the crossover operation is operated.
2. In mutation operation, a sensor selects its new parent from the sensors which is its neighbor and not in its current subtree.

The rules above ensure that any cycles aren't generated and the routing tree keeps its connectedness during GA operations. As a result, GA operations can be performed more freely and has more search space to further extend network lifetime. Since the GA operations are restricted by subtree, the proposed algorithm is named as SR-GA-MMLT (Subtree Restricted GA-MMLT).

The following symbols are employed to describe the proposed algorithm:

- n_gen: the number of generations of GA,
- n_pop: the size of population of GA,

pc: crossover probability, i.e. the probability with which an individual is selected to crossover,

pm: mutate probability, i.e. the probability with which an individual is selected to mutate,

i: an individual of GA,

tree(i): the routing tree corresponding to i,

s: the sink,

v: a sensor,

hop(v): the hops of v from the sink,

n(v, t): the set of the neighbor sensors of v in routing tree t,

s(v, t): the set of the sensors in the subtree with v as the root in routing tree t,

L_{\max} : the maximal sensor load in a routing tree.

4.1 Genetic coding and initial population

As described before, each chromosome indicates a routing tree, as shown in Fig. 1. In a chromosome, each gene indicates a sensor's parent. The method producing initial populations in GA-MMLT is adopted. In detail, all links' weights are set as 1. Then a routing tree is established with Dijkstra algorithm. Then for any sensor v, the value of hop(v) is obtained. All sensors are partitioned into different layers according to their values of hop(v) [9]. For sensor v, it randomly selects its parent from the following set [9]:

$$p(v) = \{x | x \in n(v), h(v) - h(x) = 1\} \quad (8)$$

After all sensors randomly select their parent, a routing tree (i.e. an individual) is generated. The procedure repeats to generate n_pop individuals to constitute the initial population.

4.2 Fitness function and selection

For individual i, we can construct a routing tree based on its genes. Then for sensor v, it is conceived to generate and send a data packet to the sink. Therefore, energy consumption of all sensors on the path v->...->s can be computed by equation 4 and 5. After all sensors are conceived to do this, energy expended by every sensor during the procedure, as its load, can be calculated. Then L_{\max} of the routing tree can be obtained. And the value of the fitness function of individual i is L_{\max} [9].

To eliminate inferior individuals and meanwhile reserve superior individuals, selection is implemented with the ranking selection method in which the probability an individual is chosen increases with its fitness function's value.

4.3 Crossover

The probability an individual is chosen to implement crossover is pc. Once two individuals are selected, they perform crossover to generate new offspring with the

uniform crossover operator in which each gene is randomly selected with a probability [9, 23, 24]. For two selected individuals, if gene k is selected and k_1 and k_2 is the value of gene k in i_1 and i_2 respectively, we compute the following equation:

$$\text{isCrossover} = (k_2 \notin s(k, \text{tree}(i_1))) \& k_1 \notin s(k, \text{tree}(i_2)) \quad (9)$$

Only when the value of isCrossover is true the crossover is operated. Otherwise, the crossover isn't permitted.

4.4 Mutation

The probability an individual is chosen to perform mutation is pm . If an individual is selected, its every gene is randomly selected to mutate with a probability [9, 23, 24]. If gene k of individual i is chosen to mutate, it select its new parent from the following set:

$$m(v) = \{x | x \in n(v, \text{tree}(i)) \cap x \notin s(v, \text{tree}(i))\} \quad (10)$$

This ensures that a sensor doesn't select a sensor in its subtree as its new parent.

The proposed algorithm is centralized. Every sensor informs its position to the sink once the network is deployed. After the sink receives all sensors' positions, it calculates a routing tree with the proposed algorithm and broadcasts the routing tree to the entire network before the network starts working [9]. The huge computation of GA doesn't affect the network operation because the sink has tremendous computation power and can complete the compute timely.

5 Experimental Validation

We employ OMNet++ to perform the simulations to validate the proposed algorithm. Four algorithms are compared in the simulations:

1. Power Efficient Data Gathering and Aggregation Protocol (PEDAP) [18], with which the energy consumption of a link is set as the weight of the link, and routing trees are constructed with Minimum Spanning Tree (MST);
2. Least Energy Tree (LET) [19], with which the energy consumption of a link is set as the weight of the link, and a minimum path tree with the sink as the root is constructed as the routing tree;
3. GA-MMLT (GA Based Minimal Maximal Load Tree) [9], with which all sensors are partitioned into different layers and GA operation only can be operated between two adjacent layers;
4. The proposed SR-GA-MMLT (Subtree Restricted GA-MMLT), which can perform GA operations more freely and has more search space compared to GA-MMLT.

Table 1 shows the values of the GA's main parameters. In addition, an elitist approach with which 10% of the individuals having the best fitness function values are retained to progress the next generation [9, 11, 24] is employed in SR-GA-MMLT.

Table 1. The values of the GA's main parameters

Parameters	Values
The number of generations n_gen	200
The size of population n_pop	10000
Crossover probability pc	0.9
Mutation probability pm	0.1

We have mainly compared the following two performances of the four algorithms: (1) Network lifetime; (2) Maximal sensor load, which determines network lifetime. As mentioned before, we define the time when the first sensor depletes its energy as network lifetime. The following table shows the values of the main parameters in simulations.

Table 2. The values of the main parameters in simulations

Parameters	Values
The size of the network area	400×400m ²
The number of sensors	100~200
The communications radius of sensors	80m
The initial energy of sensors	10.0Joules
The size of a data packet	1×10 ³ bits
The generating rate of data packets	1packet/second
E _{elec} in the energy model	50nJ/bit
E _{amp} in the energy model	100pJ/bit/m ²
γ in the first order radio model	2

The performances of network lifetime of four algorithms are shown in Fig. 6 and 7, in which every dot is the average value of 3 networks. As can be seen in Fig. 6, SR-GA-MMLT extends network lifetime about 360%, 159%, 31% respectively compared to PEDAP, LET and GA-MMLT on average. Fig. 7 shows that SR-GA-MMLT reduces maximal sensor load 77%, 58% and 34% respectively compared to PEDAP, LET and GA-MMLT on average. This is because PEDAP and LET don't aim to reduce maximal sensor load, and higher maximal sensor load limits network lifetime. Both GA-MMLT and SR-GA-MMLT aim at reducing maximal sensor load with GA. However, SR-GA-MMLT performs GA operations more freely and has more search space. Therefore, SR-GA-MMLT obtains lower maximal sensor load and longer network lifetime.

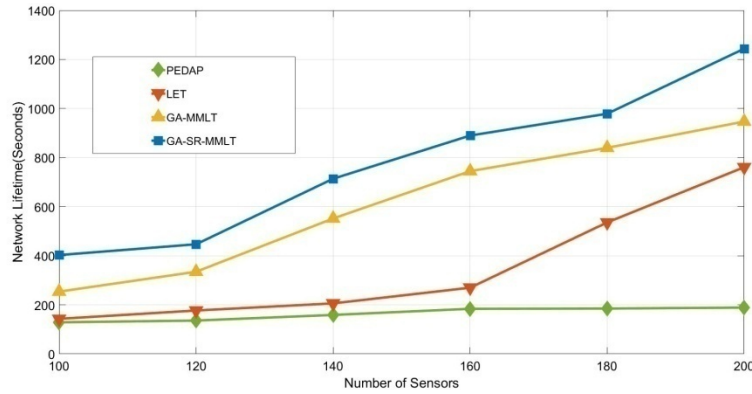


Fig. 6. Network lifetime of the four algorithms

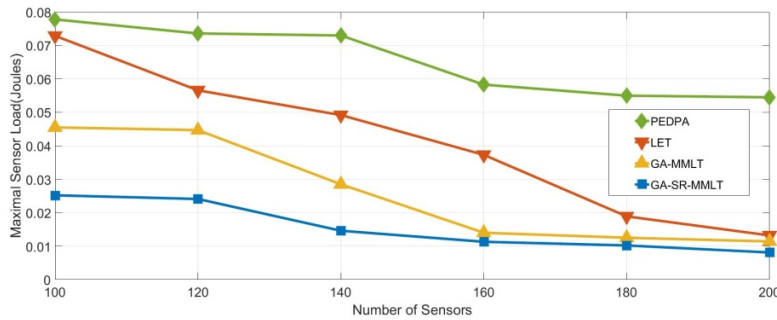


Fig. 7. Maximal sensor load of the four algorithms

6 Conclusion

In this paper, we have found that GA-MMLT limits search space and finding more optimal routing trees, and proposed a new algorithm named as GA-SR-MMLT which is based on GA-MMLT. The new proposed algorithm doesn't divide sensors into different layers or stipulate that GA operations are only implemented between adjacent layers in order to perform GA operations more freely and extend search space. And meanwhile, two rules are constituted to avoid generating cycles and keep networks connected. As a result, more optimal routing trees are obtained and network lifetime is prolonged.

7 References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey", *Computer Networks*, vol.38, no.4, pp.393–422, 2002. [https://doi.org/10.1016/S1389-1286\(01\)00302-4](https://doi.org/10.1016/S1389-1286(01)00302-4)

- [2] Garzas J. Joaquin Escudero, Calzon Carlos Bousono, and Armada Ana Garcia, “An energy-efficient adaptive modulation suitable for wireless sensor networks with SER and throughput constraints”, *Eurasip Journal on Wireless Communications and Networking*, vol.2007, no.6, pp.1-7, 2007.
- [3] R. Min., “Low power wireless sensor networks”, *The Proceedings of International Conference on VLSI Design*, vol.1, no.1, pp.5-21, 2001. <https://doi.org/10.1109/ICVD.2001.902661>
- [4] W. Li, P. Yi, Y. Wu, L. Pan, and J. Li, “A New Intrusion Detection System Based on KNN Classification Algorithm in Wireless Sensor Network”, *Journal of Electrical and Computer Engineering*, vol.2014, 2014.
- [5] K. Akkaya, M. Younis, and M. Bangad, “Sink repositioning for enhanced performance in wireless sensor networks. *Computer Networks*”, vol.2005, no.49, pp.512–534, 2005. <https://doi.org/10.1016/j.comnet.2005.01.014>
- [6] C. Ok, S. Lee, P. Mitra, and S. Kumara, “Distributed energy balanced routing for wireless sensor networks”, *Computers & Industrial Engineering*, vol.2009, no.57, pp.125–135, 2009. <https://doi.org/10.1016/j.cie.2009.01.013>
- [7] C. Vargas-Rosales, J. Mass-Sanchez, E. Ruiz-Ibarra, D. Torres-Roman, and A. Espinoza-Ruiz, “Performance Evaluation of Localization Algorithms for WSNs”, *International Journal of Distributed Sensor Networks*, vol.2015, 2015.
- [8] X. Cai, Y. Duan, and Y. He, J. Yang, and C. Li, “Bee-Sensor-C: An Energy-Efficient and Scalable Multipath Routing Protocol for Wireless Sensor Networks”, *International Journal of Distributed Sensor Networks*, vol.2015, 2015.
- [9] Z. Huang, “A Genetic Algorithm Based Minimal Maximal Load Tree Routing Algorithm in Wireless Sensor Networks”, *Journal of Electrical and Computer Engineering*, vol. 2015, 2015.
- [10] Y. Fu, Q. Guo, and C. Chen, “A-LNT: A Wireless Sensor Network Platform for Low-Power Real-Time Voice Communications”, *Journal of Electrical and Computer Engineering*, vol. 2014, 2014.
- [11] A. Bari, S. Wazed, A. Jaekel, and S.Bandyopadhyay, “A genetic algorithm based approach for energy efficient routing in two-tiered sensor networks”, *Ad Hoc Networks*, vol.2009, no.7, pp.665–676, 2009. <https://doi.org/10.1016/j.adhoc.2008.04.003>
- [12] Y. Bi, L. Sun, N. Li, “BoSS: a moving strategy for mobile sinks in wireless sensor networks”, *International Journal of Sensor Networks*, vol.2009, no.5, pp.173–184, 2009. <https://doi.org/10.1504/IJSNET.2009.026365>
- [13] Akkaya. Kemal, Younis. Mohamed, “A survey on routing protocols for wireless sensor networks”, *Ad Hoc Networks*, vol.3, no.3, pp.325-349, 2005. <https://doi.org/10.1016/j.adhoc.2003.09.010>
- [14] Aslam Nauman, Phillips William, Robertson William, et al, “A multi-criterion optimization technique for energy efficient cluster formation in wireless sensor networks”, *Information Fusion*, vol.12, no.3, pp.1847-1864, 2011.
- [15] Yu Zhenhua, Fu Xiao, Cai, Yuanli, et al, “A Reliable Energy-Efficient Multi-Level Routing Algorithm for Wireless Sensor Networks Using Fuzzy Petri Nets”, *Sensors*, vol.11, no.3, pp.3381-3400, 2011. <https://doi.org/10.3390/s110303381>
- [16] Jin Wang, Tinghuai Ma, Jinsung Cho, et al, “An Energy Efficient and Load Balancing Routing Algorithm for Wireless Sensor Networks”, *Computer Science and Information System*, vol.4, no.8, pp.991-1007, 2011. <https://doi.org/10.2298/CSIS110228052W>
- [17] Lee Sungryou, Choe Han, Park Byoungchang, et al, “LUCA: An Energy-efficient Unequal Clustering Algorithm Using Location Information for Wireless Sensor Networks”, *Wire-*

- less Personal Communications, vol.56, no.4, pp.715-731, 2011. <https://doi.org/10.1007/s11277-009-9842-9>
- [18] H. O. Tan, I. Korpeoglu, “Power efficient data gathering and aggregation in wireless sensor networks”, ACM SIGMOD, vol.32, no.4, pp.66–71, 2003. <https://doi.org/10.1145/959060.959072>
- [19] Y. Zhu, D. Shen, W. Wu, Z. Shen, and Y. Tang. “Dynamic Routing Algorithms Optimizing Lifetime of Wireless Sensor Networks”, Chinese Journal of Electronics, vol.37, no.5, pp.1041–1045, 2009.
- [20] A. G. Delevar, A. A. Baradaran, and J. Artin, “RGWSN: Presenting a genetic-based routing algorithm to reduce energy consumption in wireless sensor network”, International Journal of Computer Science Issues, vol.8, no.5, 2011.
- [21] Ioannis C, Athanasios K, Sotiris N, “Efficient data propagation strategies in wireless sensor networks using a single mobile sink”, Computer Communications, vol.31, pp.896–914, 2008. <https://doi.org/10.1016/j.comcom.2007.12.011>
- [22] Heinzelman WB, Chandrakasan AP, and Balakrishnan H, “An application-specific protocol architecture for wireless micro sensor networks”, Wireless Communications, IEEE Transactions, vol.1, no.4, pp.660–670, 2002. <https://doi.org/10.1109/TWC.2002.804190>
- [23] J. Xiong, D. T. Gao, Q. H. Shen, and S. D. Du, “Comparison of Crossover Operators in Genetic Algorithm”, Journal of Nanjing University (Natural Sciences), vol.40, no.4, 2004.
- [24] L. Davis, “Handbook of Genetic Algorithms”, 1991.

8 Author

Zhi Huang is with the School of Information Engineering, Mianyang Teachers' College, Mianyang, 621000, China.

Article submitted 23 November 2017. Resubmitted 09 January and 11 January 2018. Final acceptance 05 March 2018. Final version published as submitted by the authors.