

Full Length Research Paper

A learning automata-based algorithm using stochastic minimum spanning tree for improving life time in wireless sensor networks

Chamran Asgari^{1*}, Rohollah Rahmati Torkashvand² and Masoud Barati³

¹Department of Computer Engineering, Payame Noor University, Iran.

²Department of Computer Engineering, Islamic Azad University, Borujerd Branch, Lorestan, Iran.

³Department of Computer, Islamic Azad University, Kangavar branch, Kangavar, Iran.

Accepted 28 August, 2012

Several algorithms have already been provided for problems of data aggregation in wireless sensor networks, which somehow tried to increase networks lifetimes. In this study, we dealt with this problem using a more efficient method by taking parameters such as the distance between two sensors into account. In this paper, we presented a heuristic algorithm based on distributed learning automata with variable actions set for solving data aggregation problems within stochastic graphs where the weights of edges change with time. To aggregate data, the algorithm, in fact, creates a stochastic minimum spanning tree (SMST) in networks where variable distances of links are considered as edges, and sends data in the form of a single packet to central node after data was processed inside networks. To understand this subject better, we modeled the problem for a stochastic graph having edges with changing weights. Although this assumption that edges weights change with time makes our task difficult, the results of simulations indicate relatively optimal performance of this method.

Key words: Data aggregation, stochastic graph, learning automata, minimum spanning tree, life time.

INTRODUCTION

Wireless sensor networks consist of a large number of inexpensive sensor nodes distributed densely in the environment, having limited energy and on the other hand, consuming a great deal of energy in order to send information to central node directly. Thus, in most cases, nodes communicate with central node via their neighbors (Gupta and Kumar, 2000). On one hand, there are different paths to central node from each node, so optimal path must be selected. The frequent use of one path results in energy reduction of sensors located on that path, ultimately resulting in sensor loss. Therefore, we tried to increase networks lifetime by providing an intelligent algorithm and taking such parameters as sensor lifetime, remaining and consumption energies of

sensors and distances between sensors into account, in order to have an almost optimal data aggregation in networks. The proposed algorithm includes some steps at each of which one of possible spanning trees is created randomly.

The proposed algorithm (LA-SMST) is based on distributed learning automata, and each step of algorithm begins with selecting one of graph's nodes randomly in order to discover spanning trees and surveys of distributed learning automata using backtracking technique. Learning automaton related to chosen nodes is activated and selects one action (one edge) based on action probability vector. The edge related to this selection is added to spanning tree just formed. The

*Corresponding author. E-mail: asgari.chamran@gmail.com.

weight assigned to selected edge is added to total weight of spanning tree. To avoid forming a loop in the tree, each activated learning automaton trims its actions set by disabling actions related to already chosen edges or those edges which may form a cycle. Then, the learning automaton at the other end of selected edge is activated and selects based on of its own actions, activating the automaton located on its end (Asgari and Akbari, 2012). The process of sequential activation of learning automata (or selection of tree edges) is repeated until spanning trees are formed and / or no further action is done by current active learning automaton. Next, it performs data aggregation within middle nodes and sends the result to central node in the form of a single packet.

To create a spanning tree for data aggregating is a promising approach to reduce overhead of broadcast routing where messages are induced among minimum spanning trees. A case wireless network can be modeled as a unit disk graph $G = (V, E)$, in which nodes represent hosts and edges represent relationship between them; hosts must be in each other's transfer ranges (Clark et al., 1990; Marathe et al., 1995). Consider a network of wireless sensors located uniformly in the environment. Assume that nodes have fixed locations and identical transfer ranges. Two sensor nodes communicate directly with each other if they are in each other's transfer ranges; otherwise, they make indirect multistep communications via middle nodes. The aim of algorithm provided is to create minimum spanning trees for data aggregation in wireless sensor networks through finding a relatively optimal solution for problem of minimum spanning trees. In order to implement this approach, at first, a network of distributed learning automata is used to form this network's unit disk graph by equipping each host with a learning automaton. Then, at each step, learning automata select one of their actions randomly, considering their probability vectors until minimum spanning trees are formed. Then the minimum spanning trees formed are evaluated by random environment, and actions probability vectors from learning automata dependent on the response they receive from the environment are updated. In any iterations of this process, finally, learning automata converge to public policy of making minimum spanning trees for network graph.

This paper provides an intelligent algorithm based on distributed learning automata to aggregate data in wireless sensor networks. Each host is equipped with a learning automaton; sink node is considered as root, and then given the action probability vectors, learning automata select next action randomly from variable actions set of learning automata. This process continues until the entire network is covered and minimum spanning trees are formed. Then, the message of data aggregation is sent to all nodes from sink node in minimum spanning tree. Upon receiving the message, all nodes send their data to their parents that must wait until receiving data

from all their children. After that, parents aggregate all data and send it to their higher level parents until the aggregated data are being sent to sink node in the form of a single packet. After completing each iteration process, action probability vector of any learning automata is updated. In this study, a proposed algorithm is presented and the experiments results are demonstrated.

LITERATURE REVIEW

Many routing algorithms have been provided for sensor networks. For some of these algorithms, each node may have more than one route to sink node that one of them is selected on the basis of a series of criteria, among which the level of energy consumption along the route can be a proper criterion. Energy saving can be taken into account in two ways: (1) energy consumption is calculated for any separate routes, then the route with minimal energy consumption is chosen (Shah and Rabaey, 2002); and (2) data aggregation is based on provided learning automata, which prevents extra packets from being sent in networks by identifying sensors generating identical data and by activating sensor nodes periodically, thus saving a large amount of energy while increasing network lifetime (Esnaashari and Meybodi, 2010). A solution has been provided in Al-Karaki et al. (2009) for data aggregating and routing with internetwork aggregations in wireless sensor networks in order to maximize network lifetime by using internetwork processing techniques and data aggregation. The relationship between security and data aggregation process within wireless sensor networks has also been investigated in Ozdemir and Xiao (2009). In Soro and Heinzelman (2005), network is first clustered in order to aggregate data, and then head-clusters aggregate data from each cluster separately. A network organized into clusters with the same sizes results in unequal load distribution among head-cluster nodes. Nevertheless, Soro and Heinzelman (2005) provided a model in which clusters are of different sizes, resulting in more uniform energy distribution among head-cluster nodes and increase in network lifetime.

Furthermore, Liao et al. (2008) has offered data aggregation in wireless sensor networks by using the ant colony algorithm, which states the problem of creating data aggregation tree in wireless sensor networks for a group of source nodes to send sensed data to the single sink node. The ant colony system represents a natural method of heuristic search in determining data aggregation. Each ant discovers all possible routes to sink node and a data aggregation tree is created using accumulated pheromone. Lee and Wong (2006) also provided two different tree structures: the lifetime-preserving tree (LPT) and energy-aware spanning tree construction (E-Span) to facilitate aggregation of data

of data in wireless sensor networks. In LPT, nodes having more remaining energy are chosen as aggregation parents. The tree is restructured when one node has no long function or when a broken link is identified. E-Span is an aware energy–spanning tree algorithm in which source node with maximal remaining energy is selected as root. Other source nodes select their corresponding parents from their neighbors on the basis of such information as remaining energy and distance to root. In the report of Eskandari et al. (2009), an efficient energy–spanning tree is used to aggregate data in wireless sensor network using two parameters; energy and distance uses route energy average to balance parameters energy an distance while previously provided algorithms have selected only one of these parameters as the main one and gave sound priority to the other.

In the report of Cam et al. (2006), unlike common data aggregation methods, the energy-efficient secure pattern based data aggregation (ESPDA) avoids transmitting redundant data to head–clusters from sensor nodes in order to remove redundancy for improving application of efficient energy and bandwidth in sensor nodes. Li et al. (2010) also presented a scheme of efficient and highly accurate energy to aggregate data securely. The main idea of this is to aggregate data carefully without disclosing or reading secret information of sensors and posing considerable overhead in energy – limited sensors. In Korteweg et al. (2009) aggregation of data in wireless sensor networks is raised to balance latency and communication cost. In Korteweg et al. (2009), spanning tree- based algorithms are provided to create high convergence between data aggregation and efficient energy and low latency in wireless sensor networks. Initially, Upadhyayula and Gupta (2007) provided two algorithms for making Data aggregation enhanced convergecast (DAC) tree. The first algorithm is the kind of minimum spanning tree, and the second of individual source shortest path spanning tree, both of which are used as combined (COM) algorithm stimulator generally based on minimum spanning tree (MST) and shortest path spanning tree (SPT).

PRIMARY DEFINITIONS AND CONCEPTS

Stochastic minimum spanning tree problem

As earlier mentioned, the aim of an absolute MST algorithm is to find minimum spanning trees from graphs, assuming fixed weights of edges (Hutson and Shier, 2006). Although stochastic minimum spanning tree (SMST) algorithm concerns with graphic edges the weights of which are a stochastic variable, most scenarios assume edges weights are fixed (Ishii et al., 1981; Dhamdhare et al., 2005). But this assumption is not always true. Generally, edges of a changing network take various states (several states). Therefore, an absolute graph is not capable of modeling features of such a network really. For this reason, network topology should be modeled by a stochastic graph. As mentioned earlier, several algorithms have been proposed for solving MST problems for which

network parameters are absolute. Anyway, when the graphs are stochastic, MST is considerably difficult to find. Herein, we examine SMST problems and algorithms.

Definition 1: Graph G with stochastic weighed edges is defined by triple $\langle V, E, W \rangle$, where $V = \{V_1, \dots, V_n\}$ is edges set being a subset of $V \times V$ and $W = \{W_1, \dots, W_m\}$ is the set of weights assigned to edges set, with positive variable W_i representing the weight of edge $(e_i \in E)$ (Akbari and Meybodi, 2010).

Definition 2: Let $G \langle V, E, W \rangle$ show a stochastic weighed graph and $T = \{T_1, T_2, \dots\}$ show the set of possible spanning trees from graph G , assuming that W_{T_i} represents the expected weight of spanning tree T_i . A SMST is defined as a stochastic spanning tree with minimum expected weight where $T^* \in T$ if and only if $\min_{\forall T_i \in T} \{W_{T^*} = W_{T_i}\}$ (Akbari and Meybodi, 2010).

Learning automata

A learning automaton (LA) is an abstract model capable of doing finite actions. Each selected action is evaluated by a probable environment, the result of which is delivered to automata in the form of a positive or negative signal. Learning automata use this response to select their next action. The ultimate goal is for automatas to select the best of their actions. The best action is one maximizing the likelihood of receiving rewards from environment (Narendra and Thathachar, 1989; Thathachar and Sastry, 1997; Thathachar and Harita, 1987).

Probable environment can be expressed mathematically by triple $E \equiv \{a, \underline{B}, \underline{C}\}$ where $\underline{\alpha} \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ is the set of environment inputs and $\underline{\beta} \equiv \{\beta_1, \beta_2, \dots, \beta_r\}$ is each actions being penalized. Figure 1 shows the relationship between learning automate and environment. Given the values of $\underline{\beta}$, three different models are defined for probable environments. Whenever $\underline{\beta}$ is a two-members set of $[0, 1]$, the environment is of type P, that is-values of 0 and 1 are selected as environment outputs. In this case, $\beta_1 = 1$ means "being penalized" and $\beta_2 = 0$ means "being rewarded" (Asgari and Akbari, 2012). If $\beta(n)$ is a value bounded to $[0, 1]$, the model is of type Q; and if $\beta(n)$ is a stochastic variable within $[0, 1]$, the environment is of type S. C_i represents the probability that action α_i receives an undesirable response from environment. The values of C_i do not change in static environments while changing with time in non- static ones (Lakshminarahan and Thathachar, 1976).

Learning automatans are divided into two groups: (a) those with fixed structures and (b) those with variable structured. In this study, we made use of the variable structured. For learning automata with fixed structures, probabilities of automata actions are fixed, while for learning automata with variable structures, they are updated with each turn of iteration. Learning automatans with variable structures can be denoted by triple $\{\alpha, \beta, P, T\}$ where $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ is automata's action set; $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_r\}$ is its inputs; $P \equiv \{P_1, P_2, \dots, P_r\}$ is probability vector of each automata's action; and $\tau \equiv P(x+1) = T[\alpha(x), \beta(x), \rho(x)]$ is the learning algorithm.

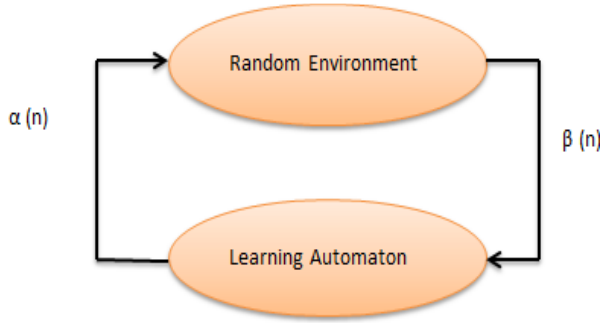


Figure 1. The relationship between learning automata and environment.

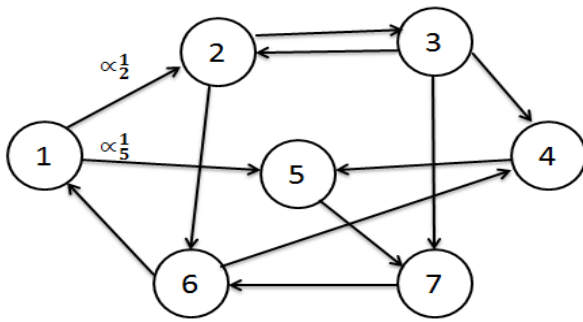


Figure 2. Network of distributed learning automata.

Automatas choose one of their actions randomly on the basis of probability vector P_i and exercise it on the environments from which they get a response. If the action selected by learning automata is

action α_i , then, automata updates its action probabilities according to Equation (1) in the case of receiving desirable response from environment, while it does this according to Equation (2) in the case of receiving undesirable one.

$$\begin{cases} P_i(n+1) = P_i(n) + a[1 - P_i(n)] \\ P_j(n+1) = (1 - a)P_j(n) \quad \forall j, j \neq i \end{cases} \quad (1)$$

$$\begin{cases} P_i(n+1) = (1 - b)P_i(n) \\ P_j(n+1) = (b/(r - 1)) + (1 - b)P_j(n) \quad \forall j, j \neq i \end{cases} \quad (2)$$

Where r is the number of automata's actions, and b is penalty parameter. The following algorithms can be available on the basis of different values considered for parameters a and b of learning:

- 1) If $a = b$, linear reward-penalty (L_{R-P}) scheme is obtained.
- 2) If the value of b is many times smaller than that of a , the resulting learning method is called liner reward epsilon scheme (L_{R-EP}).
- 3) If $b = 0$, the algorithm is called linear reward inaction scheme (L_{R-I}).

Distributed learning automata

A distributed learning automaton (DLA) (Narendra and Thathachar, 1980; Beigy and Meybodi, 2006) is a network of LAs cooperating to

solve a particular problem. Within this network of cooperating automata, only one automaton is active at a time. In DLA, the number of actions each automata is able to do is equal to the number of automata connected to that one. When an automata selects an action in the network, other automata connected to it is activated. In other words, choosing an action by an automata in this network corresponds to activation of another automata present. The model considered for DLA network is graphical, each vertex of which is an automata as shown in Figure 2.

In this graph, the presence of edge (LA_i, LA_j) means that choosing the action α_j^i by LA_i activates LA_j . The number of actions LA_k can select is denoted as $P^k = \{P_1^k, P_2^k, \dots, P_{r_k}^k\}$. Within this set, P_m^k represents probability related to action α_m^k . Selecting the action α_m^k by LA_k activates LA_m . r_k shows the number of actions LA_k is able perform.

PROPOSED STOCHASTIC MINIMUM SPANNING TREE ALGORITHM

In this paper, we proposed a heuristic algorithm called LA-SMSTA to find an optimal solution from SMST problems where edges' weights are unknown. When the weights of edges change with time, finding optimal solution from MST problem becomes too difficult. Suppose that $G(V,E,W)$ represents entries of stochastic graph, where $V = \{V_1, \dots, V_2\}$ is nodes set, $E = \{e_1, e_2, \dots, e_m\} \subseteq V \times V$ is edges set, and matrix W represents the weights assigned to edges set. In this algorithm, a network of distributed learning automata is formed by equipping each node of the graph with a learning automaton. Network results can be described with triad $\langle A, \alpha, W \rangle$ where $A = \{A_1, \dots, A_n\}$ represents a set of learning

automata; $\alpha = \{\alpha_1, \dots, \alpha_n\}$ is a set of possible actions where $\alpha_i = \{\alpha_i^1, \alpha_i^2, \dots, \alpha_i^{m_i}\}$ defines a set of actions that can be

selected by learning automata A_i (for each $\alpha_i \in \alpha$), and V_i is cardinality of action set α_i . Edge $e(i,j)$ relates either to action α_i^j

of learning automata A_i or to action α_i^j of learning automata A_j . This means that each of learning automata can select each of edges as an action. Selecting action α_i^j by automata A_i adds edge $e(i,j)$ to MST. Weight W_{ij} is the weight assigned to edge $e(i,j)$ and assumed to be a positive stochastic variable. For the proposed algorithm, all learning automata are in a passive state in the primary set.

The proposed algorithm includes some steps at each of which one of possible spanning trees is identified randomly. The algorithm is based on distributed learning automata, which surveys them by means of backtracking technique in order to discover spanning trees. Any steps of LA-SMSTA algorithm begins randomly with selecting one of graph's nodes as a sink node. Learning automata related to chosen node are activated and one action is selected based on actions probability vector. The edge related to this selection is added to spanning tree already made. The weight assigned to the chosen edge is added to total weight of spanning tree. To avoid forming a loop in a tree, each of active learning automata trims its own actions set. Then, the learning automata located at other end of chosen edge is activated, which also selects one of its own actions and activates the automata located at its end.

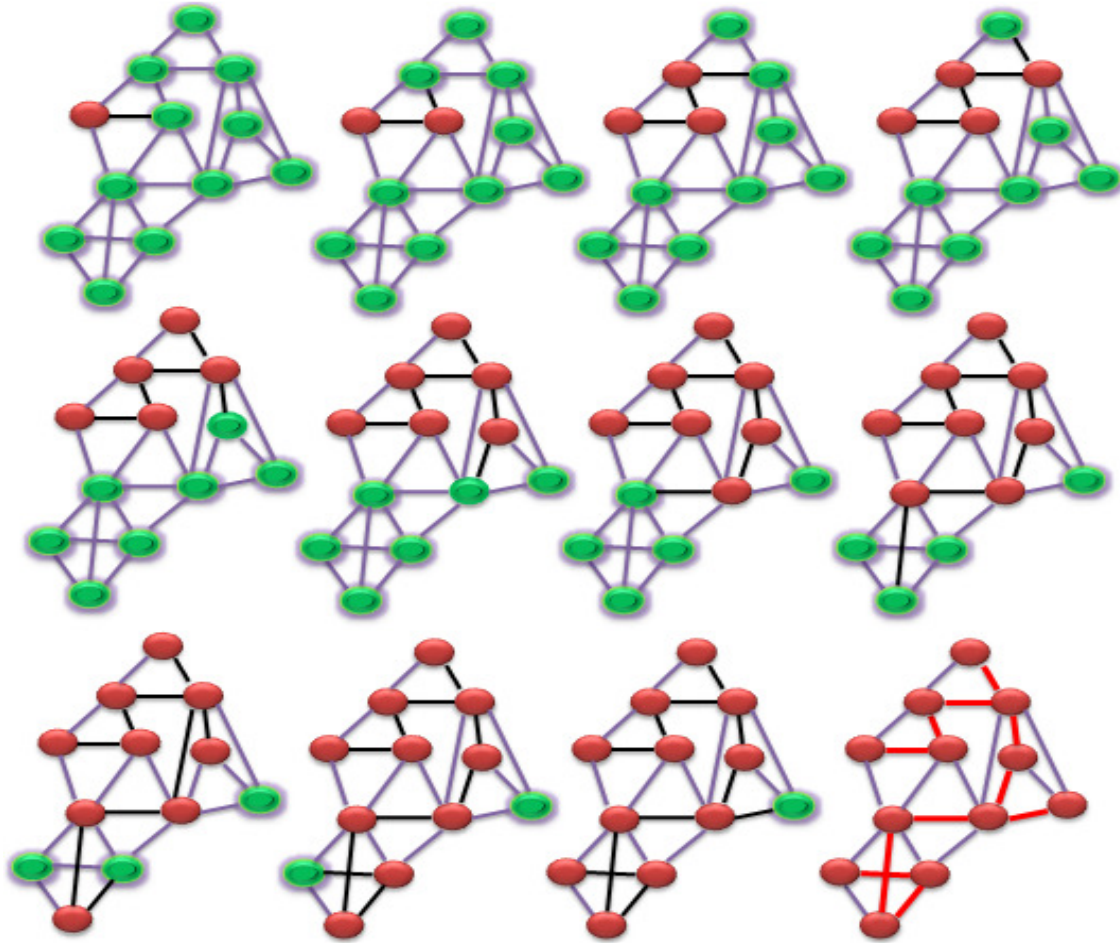


Figure 3. Steps of forming spanning trees.

The process of sequential activation is repeated from learning automata (or from selection of tree edges) until it leads to two following states: in the first state, spanning trees are formed, and in the second, current active learning automata has no action to choose. In the former, the current step is completed successfully by finding a solution for the problem of spanning trees with minimum weights (this happens when the number of selected edges $\geq n-1$, where n shows cardinality of nodes set), and in the latter, learning automata are found through backtracking process, are activated again, and actions set of automata is updated by disabling the last chosen action. Afterward, the activated automata resume the current step by selecting one of possible actions. The process of activating learning automata continues until spanning trees are formed. Then, data aggregation is performed within middle nodes and the results are sent to central node in the form of a single packet. By means of backtracking technique, each of learning automata may activate more than one of its neighbors at each step. In other words, any learning automata can select more than one action. As stated earlier, respective edge is added to spanning tree, and this task is chosen by learning automata. Also, the weight assigned to selected edge is added to total weight of spanning tree. Figure 3 shows the step of forming spanning trees. Since the weight assigned to graph edge was assumed to be a positive stochastic variable, a particular spanning tree may experience different weights. Therefore, the proposed algorithm is concerned

with the average weight of spanning trees at each step instead of the trees' own weights. To do this, at the end of the step, average weight of selected spanning tree is calculated. We assumed that spanning tree T_i was selected at step X . Average weight from spanning tree T_i to step x is calculated as follows:

$$\left\{ \begin{aligned} W_{T_i}^j &= \sum_{\forall (s,t) \in T_i} W_e^j(s,t) \\ W_e^j(s,t) &= 1 / D_{e(s,t)} \\ \bar{W}_{T_i}^x &= \frac{1}{x_i} \sum_{j=1}^{x_i} W_{T_i}^j \end{aligned} \right. \quad (3)$$

Where $W_{T_i}^j$ represents the weight of sample j^{th} from spanning tree T_i , $W_{e(s,t)}^j$ shows the weight of edge $e(s, t)$ as a part of sample j^{th} taken from spanning tree T_i , $D_{e(s,t)}$ represents the length

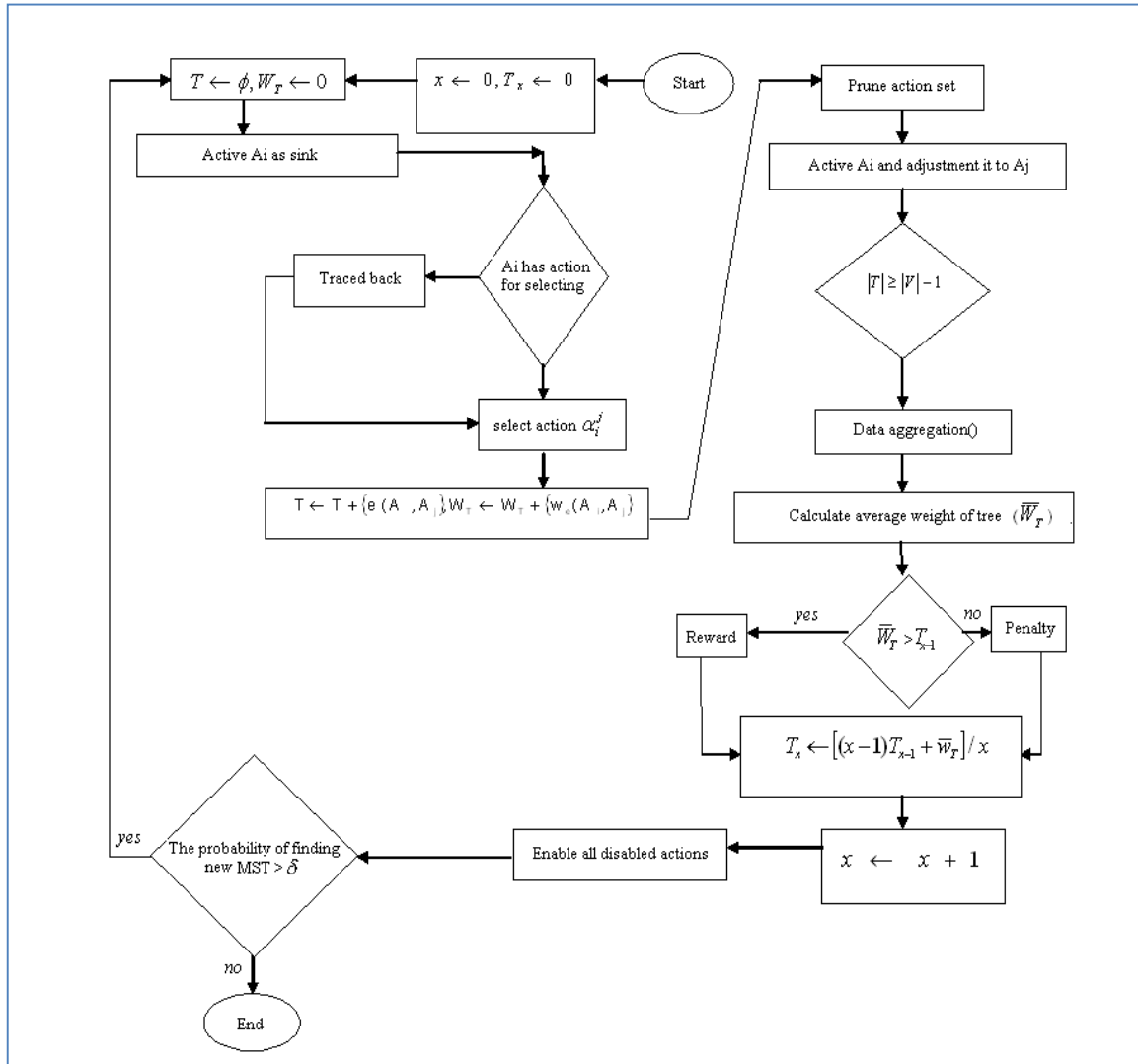


Figure 4. Flowchart of the proposed algorithm.

(distance between s and t) of edge $e(s,t)$, and $\bar{W}_{T_i}^x$ is the average weight of spanning tree T_i to step x and X_i shows the times of forming spanning tree T_i until step X .

To estimate convergence of proposed algorithm to optimal solution (minimum spanning tree), average weight of formed spanning tree is compared with dynamic threshold. At each step, T_x is compared with dynamic threshold at step $x > 1$ as follows:

$$T_x = \frac{1}{r} \sum_{i=1}^r \bar{W}_{T_i}^x \tag{4}$$

Where r shows the number of spanning trees discovered until step x . Since the weights of edges changes, a given spanning tree may be made several times, having a different weight at each time. At each step, the average weight of selected spanning tree is compared with dynamic threshold. If the average weight of selected spanning tree is bigger than the dynamic thresholds, then all learning automata reward their chosen actions, otherwise, they penalize them. Although each of learning automata updates its

action probability vector by means of learning algorithm, when learning algorithm is penalized, probability vector remains unchanged. At the final step, inactive actions need to be activated again. The process of forming spanning trees and updating action probabilities is repeated until the action probability of formed spanning trees is greater than a specific threshold called stopping threshold. Prior to stopping the algorithm, selected spanning tree is the one with minimum expected weight among all spanning trees of stochastic graph. After rewarding selected action, action probability vector must be updated again by activating all inactivated actions. Since L_{R-1} is the supporting scheme with which learning automata update their own action probability vectors, action probabilities of activated learning automata remain unchanged upon receiving penalty message. In this case, inactivated action of each learning automaton is activated again.

We can comprehend from the aforementioned that with increasing T_x , life time is increases because the relationship between life time and distances between nodes is opposite. On the other hand, when the distance between two nodes is more, the consumption energy for transferring a packet between two nodes will be more. The flowchart that depicts the proposed algorithm is shown in Figure 4.

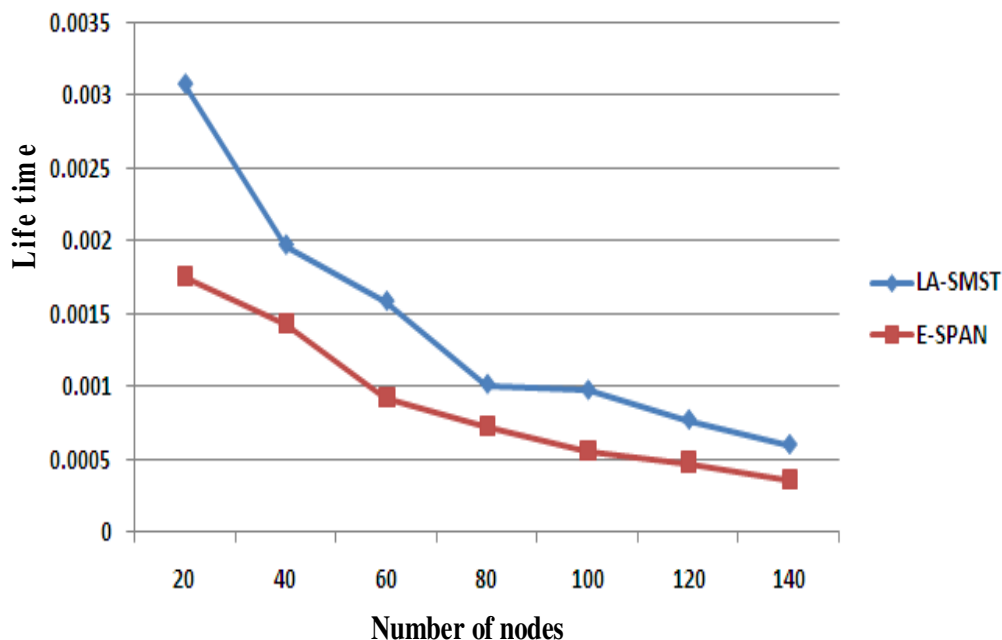


Figure 5. The relationship between life time and number of nodes.

EXPERIMENTAL RESULTS

In this paper, NS2 simulator was used to simulate wireless sensor network. Simulation was performed in a square area of $150 \times 150 \text{ m}^2$. We used L_{R-1} model for our learning automata and we assumed that the learning rate is 0.2 and the weight for each edge is allocated randomly. The maximum weights assumed were 3000. For this simulation, the threshold of SMST process and max iteration were set at 0.9 and 100, respectively. For assessing the proposed algorithm, we evaluated our simulation with respect to lifetime by increasing distances between nodes and increasing the number of nodes.

Here, for evaluating our algorithm (LA-SMST), we compared our algorithm with the proposed algorithms of Lee and Wong (2006). Lee and Wong (2006) provided two different tree structures LPT and E-Span to facilitate aggregation of data in wireless sensor networks. E-Span is an aware energy-spanning tree algorithm in which source node with maximal remaining energy is selected as root. Other source nodes select their corresponding parents from their neighbors on the basis of such information as remaining energy and distance to root.

The relationship between life time and different network scales

Here, we evaluated our simulation with respect to SMST lifetime by increasing the number of nodes. We assumed that the maximum distance between two nodes is 20 m and the number of nodes increases from 20 to

140 nodes. As show in Figure 5, the lifetime decreases when the number of nodes increases.

The relationship between life time and distances between nodes

Here, we assumed that the number of nodes is 50 and distances between nodes increase from 10 to 20 m. As show in Figure 6 with increasing distances the life is decreasing. Also, comparing our algorithm (LA-SMST) with the proposed algorithm in Lee and Wong (2006) will determine how much our method performs well.

CONCLUSION AND FUTURE WORK

In this paper, we proposed learning automata based algorithm for improving life time in wireless sensor network. Herein, we used stochastic minimum spanning tree to make a backbone. The process of making created tree was done according to the rate of distance between two nodes in the network and we tried to use route that have higher lifetime to make SMST. We also evaluated the algorithm proposed with increasing the number of nodes and distances between nodes. In addition, we compared our algorithm with other proposed algorithm and found that our algorithm always outperforms in term of the life time. The future studies will focus on increasing security of the proposed method and also fault tolerance while failing each of the sensors after forming a SMST.

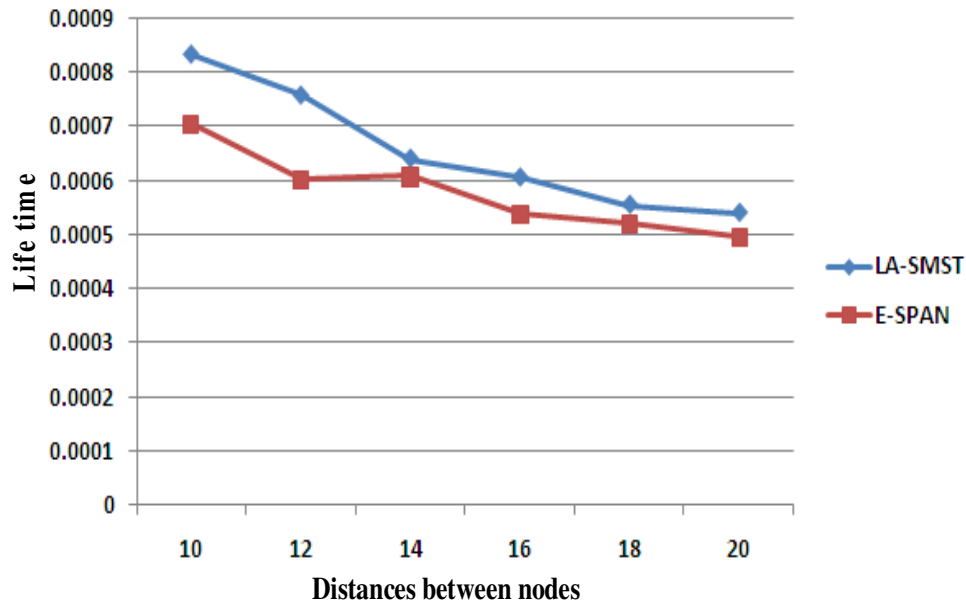


Figure 6. The relationship between life time and distances between nodes.

REERENCES

- Akbari TJ, Meybodi MR (2010). A Learning Automatabased Heuristic Algorithm for Solving the Minimum Spanning Tree Problem in Stochastic Graphs. *Comput. Netw.* 54:826-843.
- Al-Karaki JN, Ul-Mustafa R, Kamal AE (2009). Data aggregation and routing in Wireless Sensor Networks: Optimal and heuristic algorithms. *Comput. Commun.* pp. 945-960.
- Asgari CH, Akbari TJ (2012). A New Approach to the Data Aggregation in Wireless Sensor Networks. *International J. Comput. Sci. Iss. (IJCSI)* 1:328-335.
- Asgari CH, Akbari TJ (2012). A new approach for improving lifetime in wireless sensor networks based on distributed learning automata. *Int. J. Phys. Sci.* (14):2209-2219.
- Beigy H, Meybodi MR (2006). Utilizing distributed learning automata to solve stochastic shortest path problems. *Int. J. Uncertainty Fuzz. Knowl. Based Syst.* 14:591-615.
- Cam H, Ozdemir SH, Ozgur S, Prashant N (2006). Energy-efficient secure Pattern based data aggregation for wireless sensor networks. *Comput. Commun.* pp. 446-455.
- Clark BN, Colbourn CJ, Johnson DS (1990). Unit disk graphs. *Discrete Math.* 86:165-177.
- Dhamdhare K, Ravi R, Singh M (2005). *On Two-Stage Stochastic Minimum Spanning Trees*. Springer-Verlag Berlin. pp.321-334.
- Eskandari Z, Yaghmaee M (2009). Energy Efficient Spanning Tree for Data Aggregation in wireless sensor networks. *Wireless Sensor Network*, 2009. pp. 316-323.
- Eснаashari M, Meybodi MR (2010). Data Aggregation in wireless Sensor Networks using Learning Automata. *Wireless Netw.* 16:687-699.
- Gupta P, Kumar PR (2000). The capacity of wireless networks. *IEEE Transac. Inform. Theor.* 46(2):388-404.
- Hutson KR, Shier DR (2006). Minimum Spanning Trees in Networks with varying Edge Weights. *Ann. Oper. Res.* 146:3-18.
- Ishii H, Shiode S, Nishida T, Namasuya Y (1981). Stochastic Spanning Tree Problem. *Discr. Appl. Math.* 3:263-273.
- Korteweg P, Marchetti-Spaccamela A, Leen S, Andrea V (2009). Data aggregation in sensor networks: Balancing communication and delay costs. *Theor. Comput. Sci.* pp. 1346-1354.
- Lakshmivarahan S, Thathachar MAL (1976). Bounds on the convergence probabilities of learning automata. *IEEE T. Syst. Man Cy. (SMC)* 6:756-763.
- Lee WM, Wong VWS (2006). E-Span and LPT for data aggregation in wireless sensor networks. *Comput. Commun.* pp. 2506-2520.
- Li H, Lin K, Li K (2010). Energy-efficient and high-accuracy secure data aggregation in wireless sensor networks. *Comput. Commun.* 34: 591-597.
- Liao WH, Kao Y, Fan CM (2008). Data aggregation in wireless sensor networks using ant colony algorithm. *Netw. Comput. Appl.* pp. 387-401.
- Marathe MV, Breu H, Hunt III H.B, Ravi SS, Rosenkrantz DJ (1995). Simple heuristics for unit disk graphs. *Networks* 25:59-68.
- Narendra KS, Thathachar KS (1989). *Learning Automata: An Introduction*. Prentice-Hall. New York.
- Narendra KS, Thathachar MAL (1980). On the behavior of a learning automaton in a changing environment with application to telephone traffic routing, *IEEE T. Syst. Man Cy. SMC-10* 5:262-269.
- Ozdemir S, Xiao Y (2009). Secure data aggregation in Wireless Sensor Networks: A comprehensive overview. *Comput. Netw.* pp. 2022-2037.
- Shah R, Rabaey J (2002). "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks," in the Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Orlando, FL, March, 2002.
- Soro S,Heinzelman WB (2005). Prolonging the Lifetime of Wireless Sensor Networks via Unequal Clustering. *Parallel and Distributed Processing Symposium. Proceedings.* 19th IEEE International.
- Thathachar MAL, Harita BR (1987). Learning automata with changing number of actions. *IEEE T. Syst. Man Cy. (SMG)* 17:1095-1100.
- Thathachar MAL, Sastry PS (1997). A hierarchical system of learning automata that can learn the globally optimal path. *Inf. Sci.* 42:743-766.
- Upadhyayula S, Gupta SKS (2007). Spanning tree based algorithms for low latency and energy efficient data aggregation enhanced converge cast (DAC) in wireless sensor networks. *Ad Hoc Netw.* pp. 5626-648.