Relay Node Placement to Restore Connectivity in Wireless Sensor Networks

Hong Zeng[1], Zhiping Kang[2, 3*]
[1] Department of civil engineering, Chongqing Jianzhu College, Chongqing, China
[2] Key Laboratory of Ministry of Education for Dependable Service Computing in Cyber Physical Society, Chongqing University, Chongqing, China
[3] School of Software Engineering, Chongqing University, Chongqing, China
e-mail: {cqjzc_zh, kzp712}@163.com

***Abstract*—Wireless sensor networks have been used extensively in a range of applications. Due to the harsh surroundings and their inherent vulnerability, these networks suffer from simultaneous nodes failure possibly causing the network to become partitioned into multiple disjoint segments. This paper proposes a Connecting via Virtual Force and Compromise Strategy (CVFCS) for recovering from this damage using relay nodes placement. Different from most existing restoration, it considers the segments as collections of sensor nodes, and not as some representative node. Firstly, the approach uses virtual forces to adjust existing nodes of damaged deployment for increasing coverage degree. Then it finds the proper triangular Steiner points or minimum spanning tree edges to achieve the optimal connecting all segments by using relay nodes. The simulation shows that the proposed CVFCS requires less relay nodes than other algorithms under the same damaged topology.**

Keywords-connectivity recovery; relay node placement; steiner minimal tree; wireless sensor network

I. INTRODUCTION

Wireless sensor networks (WSNs) consist of spatially distributed autonomous sensor nodes to cooperatively monitor certain events and phenomena in an interesting area. Compared with the traditional network, due to their limited resources and harsh environments, nodes are prone to easily failure or damage [1]. The failures of them not only cause coverage loss of the monitored area, but also disrupt the communication of some nodes with a base-station. In these cases, the network would be split into several disjoint segments, affect the network connectivity, and even lead to immeasurable losses of network service quality. Therefore, it is crucial to restore connectivity from such damaged WSNs. The connectivity restoration problem is an attractive field of research, whose solutions generally may be classified into three categories: relocating existing nodes [2, 3], establishing k-Connected topologies [4, 5], and placing relay nodes (RNs) [6-8]. RNs are often more expensive, capable, energy and communication resources than general sensor nodes. It is usually desirable to minimize the number of RNs.

In this paper, we propose a Connecting via Virtual Force and Compromise Strategy (CVFCS) to restore connectivity by placing minimized RNs. Firstly, it uses virtual forces to pull or push nodes towards their own optimizing placement respectively, in order to achieve uniform distribution of nodes in the same segment. These movements may reduce separation distance between different segments, and improve the coverage degree by the existing nodes in the deploying area. Then it adopts Steiner minimum tree or minimum spanning tree to place the optimal RNs for connecting segments.

The main contributions of our proposed restoration strategy can be summarized as follows: (1) it considers segments as the collection of nodes in order to using more nodes to optimize connection, which is different from most of existing methods. (2) It adjusts the deployment of existing nodes using virtual force, which not only make full use of the existing nodes, but also would reduce required RNs. (3) The RNs placing strategy takes into computational complexity and numbers of RNs, which is closer to a real application environment.

II. NETWORK MODEL AND FUNDAMENTAL DEFINITION

We assume that sensor nodes are randomly deployed or placed in predetermined locations, and every node has the same sensing range R_s. Each node may communicate with other nodes in its communication range R_c, and obtain its own position through the GPS or other positioning device. RNs have mobile function, and complete position migration accurately through some running mechanisms. In the context of this paper, a segment S_i is one disjoint partition in WSN, which is a connected set of nodes v_i.

Definition 1: Segments S_i and S_j are called neighbouring segments if there is an minimum spanning tree edge (*mst-edge*) (u, v) such that $u \in S_i$ and $v \in S_j$. Nodes u and v are called as the interface nodes of segments S_i and S_j.

Definition 2: The number of RNs required to fill the gap between the interface nodes (u, v) of two segments S_i and S_j is called the weighted of *mst-edge*, denoted as $N_{mst}(S_i, S_j) = \lceil |uv|/R_c \rceil - 1$, where R_c is the communication range of nodes.

Definition 3: Three segments S_i, S_j and S_k are called triangle segment $T(i, j, k)$ if two of them are neighbouring segments.

III. CONNECTING VIA VIRTUAL FORCE AND COMPROMISE STRATEGY

The proposed CVFCS consists of two main stages: maximizing coverage and establishing connectivity. It uses virtual force of nodes to reduce the disjoint distance between different segments, and improve coverage degree of network.

Then, it places RNs to achieve the connectivity by different strategies.

A. Virtual Force Model of Nodes

Many methods have adopted virtual forces for nodes placement to improve coverage in WSNs [9-11].The virtual force model described of CVFCS focuses on nodes and nodes, nodes and the boundary of interesting area. We assume that node v_i receives virtual force F_i from others, and node v_j exerts virtual force F_{ij} to v_i:

$$F_i = \sum_{j=1, j \neq i} F_{ij} \tag{1}$$

The interaction force F_{ij} may be attractive or repulsive. The force relationship between the nodes is given as follows:

$$F_{ij} = \begin{cases} \left(w_A(d_{ij} - d_{th}), a_{ij}\right) & d_{th} < d_{ij} \le R_c \\ 0 & d_{ij} = d_{th} \text{ or } d_{ij} > R_c \\ \left(w_R(1/d_{ij} - 1/d_{th}), a_{ij} + \pi\right) & d_{ij} < d_{th} \end{cases} \tag{2}$$

where w_A and w_R are constant coefficients, and represent the quantity of attractive and repulsive forces; d_{ij} is the Euclidean distance of two nodes, and d_{th} is the distance threshold. a_{ij} is the orientation angle from node v_j to v_i.

Each node's position would be adjusted under the effect of virtual force. If the node v_i's location is (x_i, y_i), the new position $(x_i{}', y_i{}')$is calculated as following:

$$x_i{}' = \begin{cases} x_i & \left|F_{xy}\right| < F_{th} \\ x_i + \left(\frac{F_x}{\left|\sum F_{ij}\right|}\right) \times Maxstep \times e^{-\frac{1}{F_{xy}}} & \left|F_{xy}\right| \ge F_{th} \end{cases} \tag{3}$$

$$y_i{}' = \begin{cases} y_i & \left|F_{xy}\right| < F_{th} \\ y_i + \left(\frac{F_y}{\left|\sum F_{ij}\right|}\right) \times Maxstep \times e^{-\frac{1}{F_{xy}}} & \left|F_{xy}\right| \ge F_{th} \end{cases} \tag{4}$$

where F_{xy} is the resultant force exerted on the node v_i. F_x,F_y are x-coordinate and y-coordinate forces respectively. F_{th} is the threshold of the virtual force ; $Maxstep$ is the maximum distance of the nodes moving in each iteration.

B. Steiner Minimal Tree

The Steiner minimal tree (SMT) problem is usually described as following[12]: Given a set X of n points, interconnect them by a graph of shortest length, where the length is the sum of the lengths of all edges, interconnecting X . The points n are called terminals, and the points introduced are called Steiner points, which the maximum number of Steiner points is n -2. The shortest path interconnecting without introducing Steiner points is called a minimum spanning tree (MST). According to the conjecture of Pollak-Gilbertm, the ratio of the length of an SMT to the length of an MST spanning the same set is larger than $\sqrt{3}/2$.

The SMT is non-deterministic polynomial-time hard problem, which is not known whether an optimal solution can be found by using a polynomial time algorithm. When $n = 3$, it is easy to get the Steiner point. But $n = 6$, the number of Steiner tree is 5625, which is more difficulty to find the point; $n = 8$, it is 2643795, and the growth of computational complexity is astonishing. Therefore, it is an advisable choice for only discussing the triangle Steiner points in this paper.

C. RNs Placement

Assume that the triangle segment $T(i,j,k)$ is constituted by segments S_i, S_j and S_k, $u \in S_i$, $v \in S_j$, $w \in S_k$, and $|uv| \le |uw| \le |vw|$, as shown in Fig. 1. There are two ways to place RNs for connecting the vertices of $T(i,j,k)$:

(1) Connecting segments via MST. The number of RNs for connectivity restoration is $N_{mst}(S_i,S_j,S_k) = N_{mst}(S_i,S_j) + N_{mst}(S_i,S_k) = \left\lceil |uv|/R_c \right\rceil + \left\lceil |uw|/R_c \right\rceil - 2$.The locations (x_k, y_k) of the new RNs laid on (S_i, S_k) are calculating as following:

$$\begin{cases} x_k = x_u + m \cdot R_c \cdot (x_v - x_u)/|uv| \\ y_k = y_u + m \cdot R_c \cdot (y_v - y_u)/|uv| \end{cases} \tag{5}$$

where $m = 1, \dots, \left\lceil |uv|/R_c \right\rceil - 1$,and (x_u, y_u) , (x_v, y_v) are the coordinates of nodes u and v.

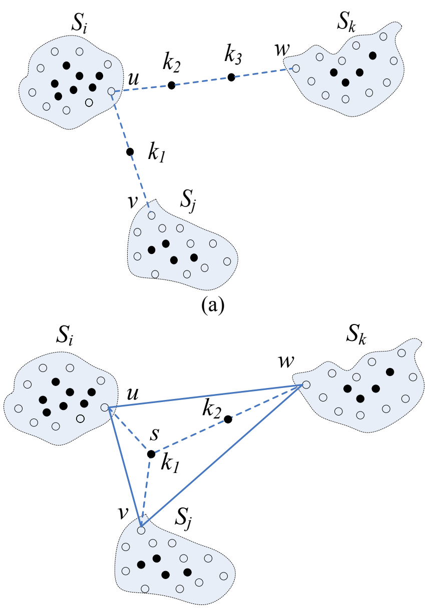


Figure 1. Different strategy for RNs placement: (a)RNs placement via MST; (b) RNs placement via SMT

(2) Connecting segments via SMT. When all interior angles of triangle are less than 120°, the Steiner point s locates inner within the triangle. The number of RNs for connectivity restoration is $N_{smt}(S_i,S_j,S_k) = N_{mst}(s,u) + N_{mst}(s,v) + N_{mst}(s,w) + 1 = \left\lceil |su|/R_c \right\rceil + \left\lceil |sv|/R_c \right\rceil + \left\lceil |sw|/R_c \right\rceil - 2$.

Set $|uv| = \bar{w}$, $|uw| = \bar{v}$, $|vw| = \bar{u}$, $|sw| = \bar{f}$, $|sv| = \bar{e}$, $|su| = \bar{d}$. The coordinates of interface nodes u, v, w are (x_u, y_u), (x_v, y_v), (x_w, y_w).The coordinate (x_s, y_s) of Steiner point s is given as following:

$$\begin{cases} x_S = \frac{(\bar{f}^2-\bar{d}^2-x_w^2+x_u^2-y_w^2+y_u^2)(y_u-y_v)-(\bar{e}^2-\bar{d}^2-x_v^2+x_u^2-y_v^2+y_u^2)(y_u-y_w)}{2[(x_u-x_w)(y_u-y_v)-(x_u-x_v)(y_u-y_w)]} \\ y_S = \frac{(\bar{f}^2-\bar{d}^2-x_w^2+x_u^2-y_w^2+y_u^2)(x_u-x_v)-(\bar{e}^2-\bar{d}^2-x_v^2+x_u^2-y_v^2+y_u^2)(x_u-x_w)}{2[(x_u-x_v)(y_u-y_w)-(x_u-x_w)(y_u-y_v)]} \end{cases} \tag{6}$$

where

$$\begin{cases} \bar{d} = (\bar{v}^2 + \bar{w}^2 - 2\bar{u}^2 + k^2)/3k \\ \bar{e} = (\bar{u}^2 + \bar{w}^2 - 2\bar{v}^2 + k^2)/3k \\ \bar{f} = (\bar{u}^2 + \bar{v}^2 - 2\bar{w}^2 + k^2)/3k \\ p = (\bar{u} + \bar{v} + \bar{w})/2 \\ S_A = \sqrt{p(p-\bar{u})(p-\bar{v})(p-\bar{w})} \\ k^2 = (\bar{u}^2 + \bar{v}^2 + \bar{w}^2)/2 + 2\sqrt{3}S_A \end{cases} \tag{7}$$

As for given three segments, N_{mst} is determined by interface point, but N_{smt} does not easily obtain. Let the number of contained nodes of three segments S_i, S_j and S_k are n_i, n_j and n_k respectively, the optimal triangle is one of all possible combination $n_i \times n_j \times n_k$ triangles.

In [6], the CIST algorithm proves that three vertexes of the optimal triangle comes from the boundary of each segment, which introduces $Gain(S_i,S_j,S_k) = N_{mst}(S_i,S_j,S_k) - N_{smt}(S_i,S_j,S_k)$ for selecting the optimal triangle with maximal gain value. It increases computational complexity due to comparing all possible triangles. Moreover, its result shows that when the number of segments is small, the improvement of CIST is not obvious. When the number is 7, 45% gains of connecting segments via SMT are 0, 28% gains is 1; when the number is 15, 18% gains of connecting segments via SMT are 0, 32% gains are 1. On the other hand, the *mst-edges* of three segments may not form one triangle, and the length of corresponding connecting via MST is less than most of other connections via SMT. Therefore, we introduces the threshold N_0 to compromise the connecting strategies for reducing the computational complexity. When $N_{mst}(S_i,S_j) \geq N_0$, it connects two segments via SMT, otherwise connecting segments via MST.

D. Algorithm Description

The CVFCS algorithm is described as follows:

The First Stage: (Input the damaged network topology)

Step 1: Calculate the exerting virtual force $\sum F_{ij}$ on each node v_i.

Step 2: Calculate the new position (x_i', y_i') under the effect of virtual force using Formula (3) and (4).

The Second Stage :(Input the adjusted deployment after using virtual force)

Step 3: Use MST algorithm to calculate $N_{mst}(S_i,S_j)$ for every neighbouring segment.

Step 4: Classify all segments into two different sets: if $N_{mst}(S_i,S_j) \geq N_0$, add the segments S_i and S_j to the set $SmtSet$; Otherwise, put them into the set $MstSet$.

Step 5: Connect all possible *mst-edge* (u,v) via MST in $MstSet$, and place RNs using Formula (5).

Step 6: Connect all segments via SMT in $SmtSet$. List all triangle segments which are formed with *mst-edges* in $SmtSet$, and calculate their $Gain(S_i,S_j,S_k)$. Select the optimal triangle with the maximal positive gain, and place RNs using Formula (5) - (7).

Step 7: Connect the remaining triangular segments via SMT, till all segments are connected. If not, connecting those segments via MST.

In Step 4, some segments may be simultaneously located in $MstSet$ and $SmtSet$, called intersection segments. For example, $N_{mst}(S_1,S_2) < N_0$,then $MstSet$ ={ (S_1,S_2) }; $N_{mst}(S_2,S_3) \geq N_0$, then $SmtSet$={(S_2,S_3)}, the segment S_2 is an intersection segment. These segments are both in two sets, and they join federating operations in Step 5 and Step 6 respectively. So they are responsible to connect the gap between $SmtSet$ and $MstSet$. Fig. 2 illustrates how the proposed CVFCS heuristic works. The dashed lines represent *mst-edges*, and the dark circles represent RNs.

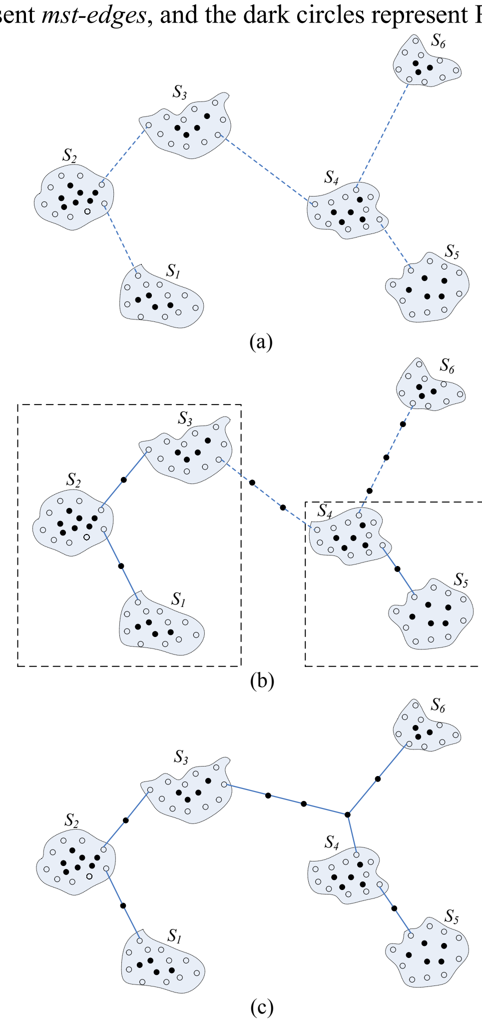


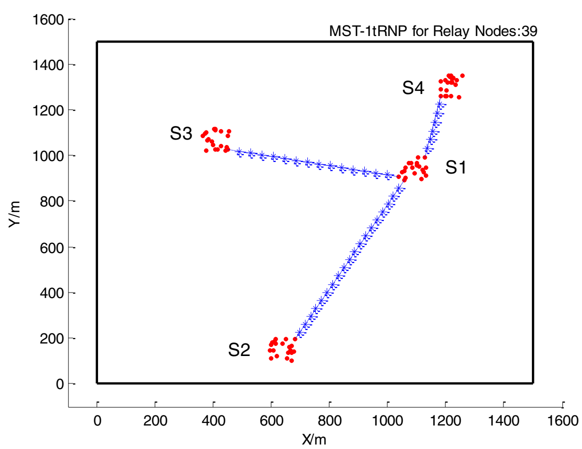
Figure 2. An illustration of how CVFCS works.

The process of using virtual force is not considered, and the adjusted topology of segments after exerting is as shown in Fig. 2 (a). Set the threshold N_0 to be 2.The proposed algorithm classifies all segments into two different sets: $MstSet$={(S_1,S_2), (S_2,S_3), (S_4,S_5)},$SmtSet$= {(S_3,S_4),

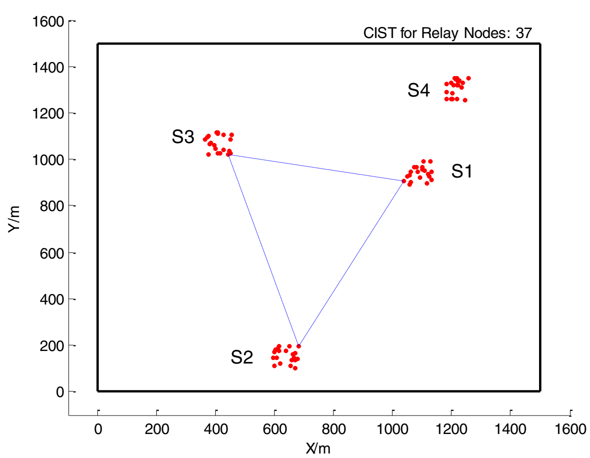
$(S_4, S_6)\}$, and the intersection segments are S_3 and S_4;The CVFCS federates (S_1, S_2), (S_2, S_3), (S_4, S_5) via MST in Fig.2(b). Then it lists all possible triangular subsets, whose edges come from all unconnected *mst-edges*. Finally, it selects the optimal triangle with the highest positive gain to federate S_3, S_4, S_6 via SMT. As shown in Fig. 2(c), the possible and optimal is only triangular $T(S_3, S_4, S_6)$.

IV. PERFORMANCE EVALUATION

The performance of CVFCS is validated through comparing with CIST [6], MST-1tRNP [7] and VFMST in the simulation using Matlab7.0. The VFMST is a variant of MST-1tRNP with additional considering using virtual force. We have randomly located 3 to 10 segments in the area (1500m×1500m). Each segment is represented as a square area with d=100m, and randomly distributed 20 nodes in the area. The parameters are set as:R_s=20m, R_c=40m, $d_{th}=\sqrt{3}R_s$, $Maxstep$=3.5, ω_a=200, ω_r=400, F_{th}=0.0001, the maximum iteration of virtual force is 50. To compare the number of RNs for different algorithms, the experiment is classified into two parts: (A). Comparing the running process of each algorithm. (B). Comparing the average required RNs using different algorithms.



(a)

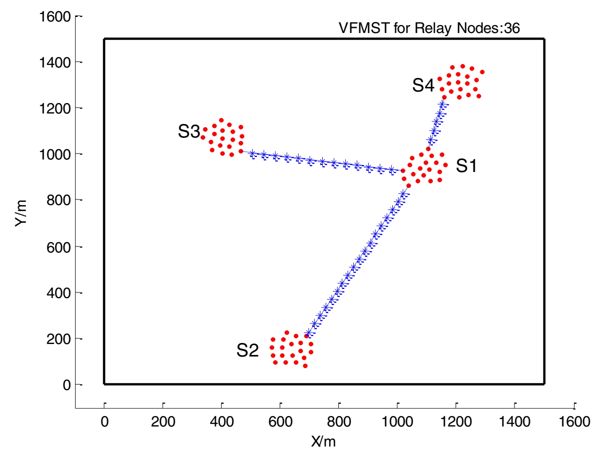


(b)

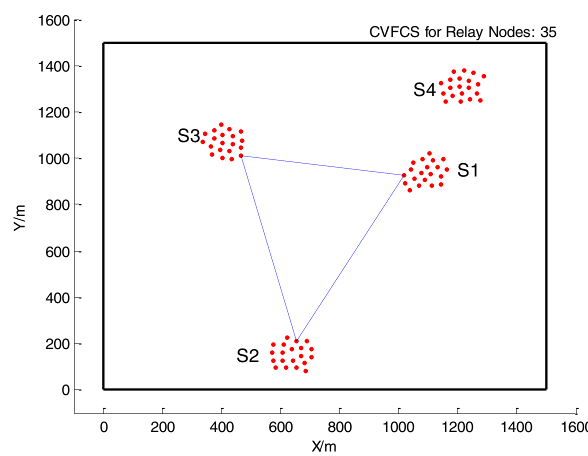
Figure 3. Restore connectivity using algorithms MST-1tRNP and CIST.

Part A: We analyze one of experiment results from different algorithms with the same deployment. There are four segments S_1, S_2, S_3, and S_4, which will be connected using RNs by algorithm CIST, MST-1tRNP, VFMST and CVFCS respectively.

In Fig. 3(a), MST-1tRNP requires total 39 RNs to federate: 19 RNs from S_1 to S_2, 14 RNs from S_1 to S_3 , 6 RNs from S_1 to S_4. In Fig. 3(b), as for CIST, there are three possible combination triangle segments using *mst-edges*, $T(S_3, S_1, S_4)$, $T(S_3, S_1, S_2)$ and $T(S_4, S_1, S_2)$. Then the algorithm calculates the gain of each triangle segment, and their maximal gain is -4, 2, -4, respectively. The negative value of gain represents that MST saves more RNs than SMT. The CIST selects $T(S_3, S_1, S_2)$ with maximal gain, and requires 37 RNs to federate: 31 RNs to connect S_1, S_2, and S_3, 6 RNs from S_1 to S_4 by *mst-edges*.



(a)



(b)

Figure 4. Restore connectivity using algorithms VFMST and CVFCS.

In Fig. 4, algorithm VFMST needs 36 RNs after exerting the virtual force: 18 RNs from S_1 to S_2, 13 RNs from S_1 to S_3 , 5 RNs from S_1 to S_4. As for CVFCS, if threshold N_0=10, there is only one triangle segment $T(S_2, S_1, S_3)$ and its gain is 1.The CVFCS requires 35 RNs: 30 RNs to federate S_2, S_1 and S_3, and 5 RNs from S_1 to S_4. If the threshold N_0=15, CVFCS algorithm is degraded to VFMST algorithm due to be no triangle segment.

From Part A, algorithm MST-1tRNP, CIST, VFMST and CVFCS require 39, 37, 36, 35 RNs, respectively. Considering two of three triangle segments are negative gains, even if the CIST adopts exerting virtual force before federating, it still would require 35 RNs. That is same as CVFCS. However, the CIST would compare 3 triangle segments, and CVFCS only requires one triangle segments in order to reduce the computing.

Part B: Compare the performance between different algorithms, when the number of segments is from 3 to 10. Each experimental result is the average of 50 independent runs. As shown in Table I, the required RNs of algorithm CVFCS and VFMST are significantly less than algorithm MST and CTST due to their using of the virtual force. With the number of segments increasing, their advantages are more obvious and satisfy the linear growth trends. With the increasing of segments from 3 to 10, the algorithm MST-1tRNP required more RNs than CVFCS respectively are 2.96, 4.50, 5.60, 6.76, 8.48, 10.86, 12.69, 14.41, the algorithm CIST required more RNs than CVFCS respectively are 2.42, 3.58, 4.88, 6.30, 7.94, 10.03, 11.69, 13.34. Meanwhile, the algorithm CVFCS is better than VFMST. When the number of segments is 3, the average required RNs are 27.46, 27.10 respectively, as for VFMST and CVFCS. When the number is 10, the required are 52.00, 51.80, respectively.

TABLE I. EXPERIMENTAL DATA OF DIFFERENT ALGORITHMS FOR RESTORING

Segments	MST-1tRNP	CIST	VFMST	CVFCS
3	30.06	29.52	27.46	27.10
4	40.30	39.38	36.22	35.80
5	44.12	43.4	38.92	38.52
6	48.06	47.6	41.48	41.3
7	52.48	51.94	44.14	44.00
8	58.02	57.19	47.32	47.16
9	62.27	61.27	49.73	49.58
10	66.21	65.14	52.00	51.80

V. CONCLUSIONS

In this paper, we bring out heuristic algorithm CVFCS to connect multiple disjoint segments using RNs placement in a destroyed WSN. It considers segments as the collection of nodes, and uses virtual force to adjust the deployment of existing nodes for reducing spacing distances of different segments. Then it places fewest RNs on the gap of segments via SMT or MST with a desirable computational complexity. The simulation results show that our proposed CVFCS is effective, and requires less RNs than algorithm MST-1tRNP, CIST, VFMST. In further research and simulations, we will use our proposed algorithms in real situations, and take quality of service, safety, and cost requirements into placing RNs between the network segments.

ACKNOWLEDGMENT

The work was supported by the key National Natural Science Foundation of China under grant 91118005.

REFERENCES

[1] Z. Kang, H. Yu, Q. Xiong, and H. Hu, "Spatial-temporal correlative fault detection in wireless sensor networks," International Journal of Distributed Sensor Networks, vol. 10, pp. 1-16, 2014.

[2] A. Abbasi, M. Younis, and U. Baroudi, "Restoring connectivity in wireless sensor-actor networks with minimal topology changes," in 2010 IEEE International Conference on Communications, 2010, pp. 1-5.

[3] A. Alfadhly, U. Baroudi, and M. Younis, "Least distance movement recovery approach for large scale wireless sensor and actor networks," in 2011 7th International Wireless Communications and Mobile Computing Conference, 2011, pp. 2058-2063.

[4] L. Sitanayah, K. N. Brown, and C. J. Sreenan, "A fault-tolerant relay placement algorithm for ensuring k vertex-disjoint shortest paths in wireless sensor networks," Ad Hoc Networks, vol. 23, pp. 145-162, 2014.

[5] X. Han, X. Cao, E. L. Lloyd, and C. C. Shen, "Fault-tolerant relay node placement in heterogeneous wireless sensor networks," IEEE Transactions on Mobile Computing, vol. 9, pp. 643-656, 2010.

[6] F. Senel and M. Younis, "Optimized connectivity restoration in a partitioned wireless sensor network," in Global Communications Conference, 2011, pp. 1-5.

[7] D. Chen, D. Z. Du, X. D. Hu, G. H. Lin, L. Wang, and G. Xue, "Approximations for steiner trees with minimum number of steiner points," Journal of Global Optimization, vol. 262, pp. 83-99, 2000.

[8] Ş. Kimençe and İ. Bekmezci, "Weighted relay node placement for wireless sensor network connectivity," Wireless Networks, vol. 20, pp. 553-562, 2014.

[9] Y. Zou and C. Krishnendu, "Sensor deployment and target localization based on virtual forces," in IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies, 2003, pp. 1293-1303.

[10] A. Dirafzoon, S. M. A. Salehizadeh, S. Emrani, and M. B. Menhaj, "Virtual force based individual particle optimization for coverage in wireless sensor networks," in CCECE 2010, 2010, pp. 1-4.

[11] F. Kribi, P. Minet, and A. Laouiti, "Redeploying mobile wireless sensor networks with virtual forces," presented at the 2009 2nd IFIP Wireless Days, Paris, France, 2009.

[12] Z. X. Yang, X. Y. Jia, J. Y. Hao, and Y. P. Gao, "Geometry-experiment algorithm for steiner minimal tree problem," Journal of Applied Mathematics, vol. 2013, pp. 507-508, 2013.