Computer Communications 81 (2016) 52–60

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Constrained node-weighted Steiner tree based algorithms for constructing a wireless sensor network to cover maximum weighted critical square grids

Bing-Hong Liu*, Ngoc-Tu Nguyen, Van-Trung Pham, Wei-Sheng Wang

Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, 415, Chien Kung Rd., Kaohsiung 80778, Taiwan, ROC

ARTICLE INFO

Article history:
Received 2 April 2015
Revised 29 July 2015
Accepted 29 July 2015
Available online 6 August 2015

Keywords:
Wireless sensor network
Coverage problem
NP-complete problem
Sensor deployment

ABSTRACT

Deploying minimum sensors to construct a wireless sensor network such that critical areas in a sensing field can be fully covered has received much attention recently. In previous studies, a sensing field is divided into square grids, and the sensors can be deployed only in the center of the grids. However, in reality, it is more practical to deploy sensors in any position in a sensing field. Moreover, the number of sensors may be limited due to a limited budget. This motivates us to study the problem of using limited sensors to construct a wireless sensor network such that the total weight of the covered critical square grids is maximized, termed the weighted-critical-square-grid coverage problem, where the critical grids are weighted by their importance. A reduction, which transforms our problem into a graph problem, termed the constrained node-weighted Steiner tree problem, is proposed and used to solve our problem. In addition, three heuristics, including the greedy algorithm (GA), the group-based algorithm (GBA), and the profit-based algorithm (PBA), are proposed for the constrained node-weighted Steiner tree problem. Simulation results show that the proposed reduction with the PBA provides better performance than the others.

1. Introduction

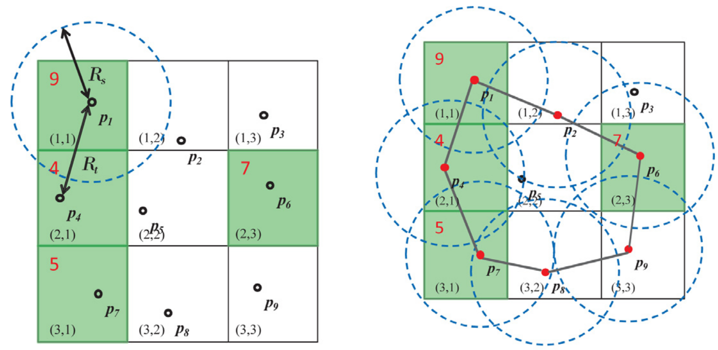
Due to the growth and advances in networking and electronic hardware techniques, wireless sensor networks have developed rapidly. In a wireless sensor network, many small sensors are deployed in a field to detect the environment and collect sensing data, such as temperature, humidity, or light data. Each sensor can process, compute, and transfer data to others in wireless sensor networks. Many applications based on the wireless sensor networks have been developed [1–6], such as medical safety protection, fire and explosion monitors, and environment monitors. Because the efficiency of the wireless sensor network is often related to the sensor deployment, the coverage problem has thus become a hot research topic, and is studied in this paper.

In the coverage problem, barrier coverage is a typical problem for applications for theft prevention and illegal intruder monitoring. For these applications, when an illegal intruder comes into a barrier coverage area, at least one sensor in the area will detect the event to ensure area security. In [7], methods are proposed to select and activate minimum sensors from sensors that are randomly deployed in a field to form barrier coverage. In [8], a sink-constructed barrier coverage method is proposed to construct virtual barrier coverage for optimizing the detection degree, detection quality, and transmission latency of a wireless sensor network. In [9,10], methods are proposed to schedule mobile sensors to construct barrier coverage in order to eliminate blind spots. In [11], the deployment for barrier coverage is studied when sensors are dropped from an aircraft along a given path. The study shows that the barrier coverage of line-based normal random offset distribution provides better performance than that of the Poisson model.

Full coverage is an important problem for applications that continuously monitor an entire area [12,13]. In [14], efficient methods are proposed to schedule sensors to be activated or inactivated to form a wireless sensor network that can cover the entire sensing field. In [15,16], a deployment-polygon-based method is proposed to achieve full coverage and k-connectivity. In addition, their proposed method has been proved to have optimal solutions for constructing wireless sensor networks under different ratios of the sensor transmission range to the sensor sensing range. In [17], the method that uses a combination of static sensors and a mobile-robot is proposed to deploy sensors such that the sensing field can be fully covered. In [18], methods are proposed to activate sensors whose total cost is minimized to fully cover the sensing field. In [19], sensor deployment

* Corresponding author. Tel.: +886 7 3814526 ext.5610; fax: +886 7 3811182.
E-mail addresses: bhliu@kuas.edu.tw, bhliu@cc.kuas.edu.tw (B.-H. Liu), kuastu2011@gmail.com (N.-T. Nguyen), pvtrung@pdu.edu.vn (V.-T. Pham), dj3662002@hotmail.com (W.-S. Wang).

http://dx.doi.org/10.1016/j.comcom.2015.07.027
0140-3664/



(a) **(b)**

Fig. 1. Example of the weighted-critical-square-grid coverage problem. (a) A sensor field divided into 9 grids of squares with length ℓ, where $R_s = \frac{\sqrt{3}}{2}\ell$, $R_t = \ell$, every grid is labeled with a pair of numbers, four critical grids are labeled with (1, 1), (2, 1), (2, 3), and (3, 1) are shown in green, the weight of each critical grid is shown in red, and hollow circles are the points that are allowed to deploy sensors. (b) A wireless sensor network constructed by 7 sensors denoted by solid circles, where an edge between two sensors represents that the two sensors can communicate with each other. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

methods are proposed to fully cover an irregular sensing field by dividing the field into many sub-regions.

Recently, a new coverage problem that studies deploying minimum sensors to cover critical areas in a sensing field has received much attention. In [20], a critical-square-grid coverage problem is proposed and shown to be NP-complete. In the problem, a sensing field is divided into square grids. The problem is to deploy minimum sensors in the center of grids to cover all critical grids. In [21], an approximation algorithm, termed the Steiner-tree-based critical grid covering algorithm (STBCGCA), is proposed for the problem. In addition, in [22], improved methods based on STBCGCA are proposed to minimize the number of deployed sensors. Note that in [21,22], the sensors must be deployed in the center of square grids.

In reality, it is more practical to deploy sensors in any position in a sensing field. In addition, sensors are often characterized by various features, such as environmental detection, intruder detection, and nuclear, biological, chemical (NBC) attack detection [23–25], due to different monitoring objectives. The prices of the sensors may be costly such that only limited sensors can be used for deployment [26–28]. That is, the critical areas in the field may not be fully covered. Therefore, critical areas must be weighted by their importance. The more important a critical area, the higher the weight of the area. For example, in a wilderness ecological observation network [21,29], the nests of animals are more important than their foraging areas. This motivates us to study a coverage problem, termed the weighted-critical-square-grid coverage problem. In the problem, the sensing field is divided into weighted square grids. Although the number of sensors and the locations of points that are allowed to deploy sensors are given, the problem is to find a connected wireless sensor network such that the total weight of the covered critical grids is maximized. In the following sections, Section 2 illustrates the problem definition and its hardness. A reduction transformed from the weighted-critical-square-grid coverage problem into a general graph problem, termed the constrained node-weighted Steiner tree problem, is proposed in Section 3 to solve the coverage problem. In addition, three centralized heuristics, termed the greedy algorithm (GA), the group-based algorithm (GBA), and the profit-based algorithm (PBA), for the constrained node-weighted Steiner tree problem are proposed in Section 4. The simulations are discussed in Section 5 to show the performance of our proposed methods. Section 6 concludes the paper.

2. Problem definition and its hardness

We first introduce our problem, termed the weighted-critical-square-grid coverage problem, in Section 2.1. In Section 2.2, we discuss the hardness of the weighted-critical-square-grid coverage problem.

2.1. The weighted-critical-square-grid coverage problem

In this paper, the sensing model in the wireless sensor network is assumed to be a binary sensor model [18,22,30], in which the probability of detecting an event by a sensor u is 1 if the event is within u's sensing range R_s; otherwise, the probability is 0. In addition, the communication model is assumed to be a unit disk graph model [31], in which a sensor u can receive messages sent from sensor v if u is within the transmission range R_t of v. Let *Field* denote a sensing field, which is divided into grids of squares that have length ℓ. In *Field*, the set of points that are allowed to deploy sensors are denoted by *Location*. In addition, the set of grids that belong to critical areas are denoted by a weighted set *Critical*, where every grid $c \in Critical$ has a weight $w(c) \in Z^+$. Hereafter, grids in *Critical* are called critical grids. Here, a critical grid is said to be covered by a sensor v deployed on a point in *Location* if the grid is fully within v's sensing range [32]. Our problem, termed the weighted-critical-square-grid coverage problem, is the problem of finding a connected wireless sensor network W in *Field*, with at most n sensors, to cover critical grids in *Critical* of the maximum total weight, while the instance, containing R_s, R_t, n, *Field*, *Critical*, and *Location*, is given. The weighted-critical-square-grid coverage problem is illustrated as follows:

INSTANCE: R_s, R_t, n, *Field*, *Critical*, *Location*, and a positive integer k.

QUESTION: Does there exist a connected wireless sensor network W in *Field*, with at most n sensors deployed on the points in *Location*, such that W covers critical grids in *Critical* with total weight no less than k?

Take Fig. 1, for example. Fig. 1a shows an instance of the weighted-critical-square-grid coverage problem, containing R_s, R_t, *Field*, *Critical*, *Location*, where $R_s = \frac{\sqrt{3}}{2}\ell$, $R_t = \ell$, *Field* is the sensor field, $Critical = \{(1, 1), (2, 1), (2, 3), (3, 1)\}$, $w((1, 1)) = 9$, $w((2, 1)) = 4$, $w((2, 3)) = 7$, $w((3, 1)) = 5$, $Location = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9\}$. Here let $n = 7$ in the instance. Fig. 1b shows a connected wireless sensor network W that has 7 sensors deployed on points p_1, p_2, p_4, p_6, p_7, p_8, and p_9, respectively. In addition, the total weight of the critical grids covered by W is 25.

2.2. The hardness of the weighted-critical-square-grid coverage problem

We show that the weighted-critical-square-grid coverage problem is NP-complete here. Because the problem of finding a minimum size connected wireless sensor network to fully cover all critical

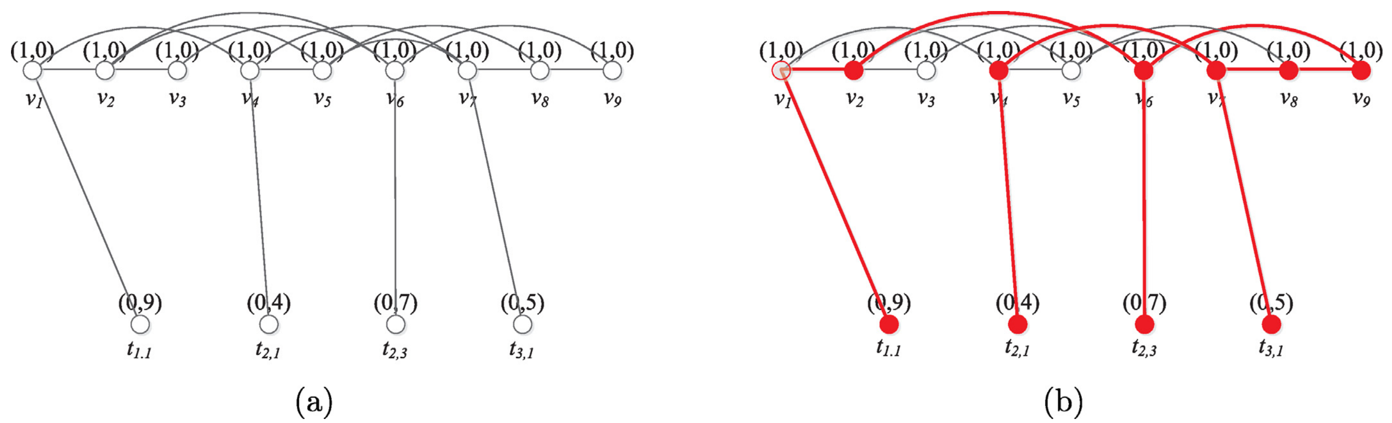


Fig. 2. Example of the constrained node-weighted Steiner tree problem. (a) An undirected weighted graph $G(V_G, E_G, w_c, w_p)$, where there are 13 nodes in V_G, $t_{1,1}$, $t_{2,1}$, $t_{2,3}$, and $t_{3,1}$ are terminal nodes, $w_c(v_1) = w_c(v_2) = w_c(v_3) = w_c(v_4) = w_c(v_5) = w_c(v_6) = w_c(v_7) = w_c(v_8) = w_c(v_9) = 1$, $w_p(t_{1,1}) = 9$, $w_p(t_{2,1}) = 4$, $w_p(t_{2,3}) = 7$, and $w_p(t_{3,1}) = 5$. (b) A tree $T(V_T, E_T)$ in G, where $V_T = \{v_1, v_2, v_4, v_6, v_7, v_8, v_9, t_{1,1}, t_{2,1}, t_{2,3}, t_{3,1}\}$ and $E_T = \{(v_1, v_2), (v_2, v_6), (v_6, v_9), (v_4, v_7), (v_7, v_8), (v_8, v_9), (v_1, t_{1,1}), (v_4, t_{2,1}), (v_6, t_{2,3}), (v_7, t_{3,1})\}$.

grids [20,22], termed the critical-square-grid coverage problem, is similar to the weighted-critical-square-grid coverage problem, we use it to show the hardness of that problem in Theorem 1. In this problem [20,22], a sensor field, denoted by F, is divided into grids of squares that have length ℓ'. The field is allowed to deploy sensors on grid points each located on the center of a grid. The set of grids in F required to be fully covered by sensors is denoted by C. Let the sensing and transmission ranges of a sensor be R_s' and R_t', respectively. The critical-square-grid coverage problem is illustrated as follows:

INSTANCE: R_s', R_t', F, C, and a positive integer n'.

QUESTION: Does there exist a connected wireless sensor network W, with no more than n' sensors deployed on the grid points in *Field*, such that W fully covers all critical grids in C?

Theorem 1. *The weighted-critical-square-grid coverage problem is NP-complete.*

Proof. Because the weighted-critical-square-grid coverage problem clearly belongs to the NP class, it suffices to show that the instance of the critical-square-grid coverage problem is solvable if, and only if, the instance of the weighted-critical-square-grid coverage problem is solvable. Given any instance of the critical-square-grid coverage problem, including R_s', R_t', F, C, and n', the instance of the weighted-critical-square-grid coverage problem is created as follows: $R_s = R_s'$, $R_t = R_t'$, $n = n'$, $Field = F$, *Location* is the set of all grid points in F, $Critical = C$ in which for every $c \in Critical$ $w(c) = 1$, and k is the number of critical grids in *Critical*. It is obvious that the reduction can be performed in polynomial time. In addition, it is clear that the input instance is solvable if, and only if, the output instance is solvable. Because the critical-square-grid coverage problem [20,22] is NP-complete, this completes the proof. □

3. A reduction in the weighted-critical-square-grid coverage problem

Our idea for solving the weighted-critical-square-grid coverage problem is to transform the problem into a more general graph problem, termed the constrained node-weighted Steiner tree problem. Once a solution to the constrained node-weighted Steiner tree problem is obtained, the inverse transform is used to get the solution to the weighted-critical-square-grid coverage problem. The constrained node-weighted Steiner tree problem and its hardness are introduced in Section 3.1. In Section 3.2, the reduction in the weighted-critical-square-grid coverage problem is described.

3.1. The constrained node-weighted Steiner problem and its hardness

Let $G(V_G, E_G, w_c, w_p)$ be an undirected weighted graph, where V_G (or E_G) is a set of nodes (or edges), every node $v \in V_G$ has a non-negative cost value and a non-negative profit value, and $w_c(v)$ (or $w_p(v)$) denotes the cost (or profit) value of node v for $v \in V_G$. In G, several nodes are special nodes, termed terminal nodes. Let S be a set of terminal nodes in G. Given $G(V_G, E_G, w_c, w_p)$, S, and a positive integer *budget*, the constrained node-weighted Steiner tree problem is finding a tree $T(V_T, E_T)$ in G such that v is a leaf in T for any $v \in V_T \cap S$, the total cost of nodes in V_T is not greater than *budget*, and the total profit of nodes in V_T is maximum, which is illustrated as follows:

INSTANCE: $G(V_G, E_G, w_c, w_p)$, S, *budget*, and a positive integer x.

QUESTION: Does there exist a tree $T(V_T, E_T)$ in G such that v is a leaf in T for any $v \in V_T \cap S$, the total cost of nodes in V_T is not greater than *budget*, and the total profit of nodes in V_T is not less than x?

Take Fig. 2, for example. Fig. 2a shows an instance of the constrained node-weighted Steiner tree problem, containing $G(V_G, E_G, w_c, w_p)$ and S, where $V_G = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, t_{1,1}, t_{2,1}, t_{2,3}, t_{3,1}\}$, $E_G = \{(v_1, v_2), (v_1, v_4), (v_2, v_3), (v_2, v_5), (v_2, v_6), (v_3, v_6), (v_4, v_5), (v_4, v_7), (v_5, v_7), (v_5, v_8), (v_6, v_9), (v_7, v_8), (v_8, v_9), (v_1, t_{1,1}), (v_4, t_{2,1}), (v_6, t_{2,3}), (v_7, t_{3,1})\}$, $S = \{t_{1,1}, t_{2,1}, t_{2,3}, t_{3,1}\}$. Here let $budget = 7$ in the instance. Fig. 2b shows a tree $T(V_T, E_T)$ in G whose total cost and total profit are 7 and 25, respectively, where $V_T = \{v_1, v_2, v_4, v_6, v_7, v_8, v_9, t_{1,1}, t_{2,1}, t_{2,3}, t_{3,1}\}$ and $E_T = \{(v_1, v_2), (v_2, v_6), (v_6, v_9), (v_4, v_7), (v_7, v_8), (v_8, v_9), (v_1, t_{1,1}), (v_4, t_{2,1}), (v_6, t_{2,3}), (v_7, t_{3,1})\}$. In addition, v is a leaf in T for any $v \in V_T \cap S$.

Next, we use the node-weighted Steiner tree with a budget problem in [33] to show the hardness of the constrained node-weighted Steiner tree problem in Theorem 2. Let $G'(V_{G'}, E_{G'}, w_{c'}, w_{p'})$ be an undirected weighted graph, where $V_{G'}$ (or $E_{G'}$) is a set of nodes (or edges) and $w_{c'}(v)$ (or $w_{p'}(v)$) denotes the cost (or profit) value of node v for $v \in V_{G'}$. In the node-weighted Steiner tree with a budget problem, given $G'(V_{G'}, E_{G'}, w_{c'}, w_{p'})$ and a positive integer b, the problem is finding a tree $T'(V_{T'}, E_{T'})$ in G' such that the total cost of nodes in $V_{T'}$ is not greater than b and the total profit of nodes in $V_{T'}$ is maximum, which is illustrated as follows:

INSTANCE: $G'(V_{G'}, E_{G'}, w_{c'}, w_{p'})$, b, and a positive integer x'.

QUESTION: Does there exist a tree $T'(V_{T'}, E_{T'})$ in G' such that the total cost of nodes in $V_{T'}$ is not greater than b and the total profit of nodes in $V_{T'}$ is not less than x'?

Theorem 2. *The constrained node-weighted Steiner tree problem is NP-complete.*

Proof. Because it is clear that the constrained node-weighted Steiner tree problem belongs to the NP class, it suffices to show that the instance of the node-weighted Steiner tree with a budget problem is solvable if, and only if, the instance of the constrained node-weighted Steiner tree problem is solvable. Given any instance of the node-weighted Steiner tree with a budget problem, including $G'(V_{G'}, E_{G'}, w_{c'}, w_{p'})$, b, and x'. the instance of the constrained node-weighted Steiner tree problem is created as follows: $G(V_G, E_G, w_c, w_p) = G'(V_{G'}, E_{G'}, w_{c'}, w_{p'})$, $S = \emptyset$, $budget = b$, and $x = x'$. It is obvious that the reduction can be performed in polynomial time. In addition, it is clear that the input instance is solvable if, and only if, the output instance is solvable. Because the node-weighted Steiner tree with a budget problem [33] is NP-complete, this completes the proof. □

3.2. A reduction

In the reduction, our idea is to first transform the weighted-critical-square-grid coverage problem into the constrained node-weighted Steiner tree problem. In the transformation, while given the instance of the weighted-critical-square-grid coverage problem, including R_s, R_t, n, *Field*, *Critical*, and *Location*, the instance of the constrained node-weighted Steiner tree problem, including a graph $G(V_G, E_G, w_c, w_p)$, a set of terminal nodes S, and *budget*, is constructed. Once a solution to the constrained node-weighted Steiner tree problem is obtained, the solution is used to select the points that are allowed to deploy sensors for the weighted-critical-square-grid coverage problem.

In the construction of G, each critical grid $c \in Critical$ is denoted by a terminal node whose cost value and profit value are equal to 0 and the weight of c, that is, $w(c)$, respectively; each point $p \in Location$ is denoted by a non-terminal node whose cost value and profit value are equal to 1 and 0, respectively; there is a link between a non-terminal node and a terminal node if the sensor deployed on the point denoted by the non-terminal node can cover the critical grid denoted by the terminal node; there is a link between two non-terminal nodes if two sensors deployed on the points denoted by the two non-terminal nodes can communicate with each other. In addition, S is set to the set of all terminal nodes in G, and *budget* is set to n. Based on G, S, and *budget*, we can apply the algorithm proposed in Section 4 to find a tree $T(V_T, E_T)$ for the constrained node-weighted Steiner tree problem. Finally, we select the points that are denoted by the non-terminal nodes in T to deploy sensors. Given R_s, R_t, n, *Field*, *Critical*, and *Location*, the reduction, which constructs a set of points, termed *DEP*, for deploying sensors, is described in detail as the following three steps:

(**1**) **Construction of** $G(V_G, E_G, w_c, w_p)$, a set of terminal nodes S, and *budget***:** Let V_1 be the set of nodes $t_{i,j}$ for all critical grid $(i, j) \in Critical$, and let V_2 be the set of nodes v_x for all points p_x in *Location*. The cost value and the profit value of each node $t_{i,j}$ in V_1 are assigned 0 and $w((i, j))$, respectively; the cost value and the profit value of each node v_x in V_2 are assigned 1 and 0, respectively. Let $V_G = V_1 \cup V_2$. Let E_1 be the set of edges $(v_x, t_{i,j})$ for all points p_x in *Location* and critical grid $(i, j) \in Critical$, such that the sensor deployed on point p_x covers the critical grid (i, j). Let E_2 be the set of edges (v_x, v_y) for all points p_x and p_y in *Location*, such that the distance between p_x and p_y is not greater than R_t. Then, $E_G = E_1 \cup E_2$. In addition, S is set to V_1, and *budget* is set to n.

(**2**) **Construction of constrained node-weighted steiner tree *T*:** Once G, S, and *budget* are obtained, the algorithms proposed in Section 4 can be applied to construct a constrained node-weighted Steiner tree $T(V_T, E_T)$ in G.

(**3**) **Formation of set *DEP*:** *DEP* is the set of points p_x for all $v_x \in T$.

Take Fig. 2, for example. The first step is constructing the graph G, as shown in Fig. 2a, based on the instance of Fig. 1a. In constructing G, $t_{2,3}$ with $w_c(t_{2,3}) = 0$ and $w_p(t_{2,3}) = 7$ is in V_1 because critical grid $(2, 3)$ is in *Critical*; v_6 with $w_c(v_6) = 1$ and $w_p(v_6) = 0$ is in V_2 because point p_6 is in *Location*. In addition, $(v_6, t_{2,3})$ is in E_1 because the sensor deployed on point p_6 can cover the critical grid $(2, 3)$; (v_3, v_6) is in E_2 because the distance between p_3 and p_6 is smaller than R_t. The second step is constructing a constrained node-weighted Steiner tree T, such as the tree in Fig. 2b. Note that $t_{1,1}$, $t_{2,1}$, $t_{2,3}$, and $t_{3,1}$ are leaves in T. Finally, we have $DEP = \{p_1, p_2, p_4, p_6, p_7, p_8, p_9\}$ because v_1, v_2, v_4, v_6, v_7, v_8, and v_9 are in T.

3.3. Analysis of the reduction

In the following, Theorem 3 shows that sensors deployed on the points selected by the proposed reduction method can form a connected wireless sensor network. In addition, Theorem 4 shows that by our reduction method, the instance of the weighted-critical-square-grid coverage problem has an optimal solution if the corresponding instance of the constrained node-weighted Steiner tree problem has an optimal solution.

Theorem 3. *Sensors deployed on the points in DEP obtained by the reduction method form a connected wireless sensor network.*

Proof. Because each terminal node in the constrained node-weighted Steiner tree $T(V_T, E_T)$ constructed by the reduction method is a leaf, the tree after removing all terminal nodes, termed $T'(V_{T'}, E_{T'})$, is still a tree. We have that T' is connected and $V_{T'} \cap S = \emptyset$. In addition, because a link existing between two non-terminal nodes implies that two sensors deployed on the points denoted by the two non-terminal nodes can communicate with each other, we have that the network formed by the sensors deployed on the points p_x for all $v_x \in T'$ is connected, and thus, this completes the proof. □

Theorem 4. *The weighted-critical-square-grid coverage problem has an optimal solution obtained by our reduction method if the constrained node-weighted Steiner tree problem has an optimal solution.*

Proof. Given any instance I of the weighted-critical-square-grid coverage problem, we can construct the corresponding instance I' of the constrained node-weighted Steiner tree problem with the reduction method. Let T_{OPT} denote the optimal solution to the instance I'; let $DEP_{T_{OPT}}$ denote the solution to the instance I that is obtained by the reduction method based on T_{OPT}. Because T_{OPT} is a constrained node-weighted Steiner tree whose terminal nodes are leaves, and the profit value of each terminal node is 1 in G, the total profit of nodes in T_{OPT}, denoted by $w_p(T_{OPT})$, is equal to the total weight of the critical grids covered by the sensors deployed on the points in $DEP_{T_{OPT}}$, denoted by $w(DEP_{T_{OPT}})$. It suffices to show that $DEP_{T_{OPT}}$ is an optimal solution.

Assume that $DEP_{T_{OPT}}$ is not optimal. We have that $w(DEP_{OPT}) > w(DEP_{T_{OPT}})$, where DEP_{OPT} denotes the optimal solution to the instance I. We can construct a constrained node-weighted Steiner tree, $T_{DEP_{OPT}}$, based on DEP_{OPT} as the following three steps. In the first step, because the sensors deployed on the points in DEP_{OPT} can form a connected wireless sensor network, a spanning tree T_1 in G is constructed by spanning v_x for all $p_x \in DEP_{OPT}$. Because the sensors deployed on the points in DEP_{OPT} can cover a set of critical grids, called C, a critical grid in C always can be covered by a sensor deployed on a point in DEP_{OPT}. In the second step, for every $(i, j) \in C$, an edge $(v_x, t_{i,j})$ is added into T_1 in G such that p_x is in DEP_{OPT}. Finally, $T_{DEP_{OPT}}$ is the tree T_1. It is easy to verify that $T_{DEP_{OPT}}$ is a constrained node-weighted Steiner tree. It is clear that $w_c(T_{DEP_{OPT}}) \le budget$ because the cost value of every non-terminal (or terminal) node is 1 (or 0) in G, where $w_c(T_{DEP_{OPT}})$ denotes the total cost of nodes in $T_{DEP_{OPT}}$. In addition, it is also clear that $w_p(T_{DEP_{OPT}}) = w(DEP_{OPT})$ because the profit value of every non-terminal node is 0 in G, where $w_p(T_{DEP_{OPT}})$ denotes the total profit of nodes in $T_{DEP_{OPT}}$. Because $w(DEP_{OPT}) > w(DEP_{T_{OPT}})$, $w_p(T_{DEP_{OPT}}) = w(DEP_{OPT})$, and $w_p(T_{OPT}) = w(DEP_{T_{OPT}})$, we have that $w_p(T_{DEP_{OPT}}) > w_p(T_{OPT})$. This implies that the constrained node-weighted Steiner tree $T_{DEP_{OPT}}$, which is a solution to the instance I', has a higher profit

Algorithm 1 Greedy Algorithm ($G(V_G, E_G, w_c, w_p)$, $budget$).

```
Let src be the terminal node u with a highest profitvalue in V_G
Let T(V_T, E_T) be a tree with a node src
Let tot_cost be the cost of src
while tot_cost < budget do
    Let W be the set of the terminal nodes in V_G − V_T with highest profit values
    if W = ∅ then
        return T
    end if
    while W ≠ ∅ do
        Let dst be the node in W that has a minimum hop distance to src in G
        Let P be the shortest path from src to dst in G such that the nodes on the shortest path, excluding the end-points, are non-terminal nodes
        Let add be the total cost of the nodes on P − V_T
        if tot_cost + add ≤ budget then
            Add P into T
            tot_cost ← tot_cost + add
            W ← W − {dst}
        else
            return T
        end if
    end while
    src ← dst
end while
return T
```

value than T_{OPT}, which constitutes a contradiction. This completes the proof. □

4. Algorithms for the constrained node-weighted steiner tree problem

Because the constrained node-weighted Steiner tree problem is NP-complete as shown in Theorem 2, we propose three heuristics for the problem. Here, the greedy algorithm (GA), the group-based algorithm (GBA), and the profit-based algorithm (PBA) are presented, respectively, in Sections 4.1–4.3.

4.1. Greedy algorithm

In the greedy algorithm, our idea is first to select a node $u \in V_G$ with the highest profit value. Let u be a tree $T(V_T, E_T)$ by itself, and src be set to u. Then we iteratively merge the tree with other terminal nodes that have next higher profit values if the total cost of the nodes that are merged or included in the tree does not exceed $budget$. Let dst be set to the terminal node that has the next higher profit value and the minimum hop distance to src in G. Also let $P = (v_1, v_2, \ldots, v_n)$ be the shortest path from src to dst in G such that the nodes on the shortest path, excluding the end-points, are non-terminal nodes, where $v_1 = src$ and $v_n = dst$. In each iteration, the shortest path P is merged in T, that is, $V_T = V_T \cup \{v_1, v_2, \ldots, v_n\}$, and $E_T = E_T \cup \{(v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n)\}$, if the total cost of the nodes in $V_T \cup \{v_1, v_2, \ldots, v_n\}$ does not exceed $budget$. src is set to dst if the terminal nodes with the same profit values as dst are all merged in the tree T. Algorithm 1 shows the greedy algorithm in detail. In addition, Theorem 5 shows the time complexity of the greedy algorithm.

Theorem 5. *The time complexity of the greedy algorithm is bounded in $O(mn^2)$, where n and m denote the numbers of nodes and terminal nodes in G, respectively.*

Proof. It is clear that $m \le n$ because n and m denote the numbers of nodes and terminal nodes in G, respectively. In addition, we know that the time complexity of finding the shortest path in an undirected graph is $O(|E_G|)$ [34], where $|E_G|$ denotes the number of edges in E_G. Because $|E_G| \le \frac{n(n-1)}{2}$ and at most m terminal nodes are merged in the tree T, the time complexity of the greedy algorithm is bounded in $O(mn^2)$, which completes the proof. □

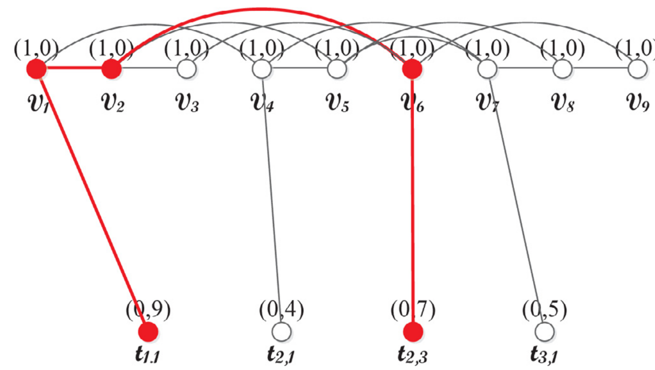


Fig. 3. A constrained node-weighted Steiner tree $T(V_T, E_T)$ constructed by the greedy algorithm in the graph G shown in Fig. 2a, where $V_T = \{v_1, v_2, v_6, t_{1,1}, t_{2,3}\}$, and $E_T = \{(v_1, t_{1,1}), (v_1, v_2), (v_2, v_6), (v_6, t_{2,3})\}$.

Take Fig. 3, for example. Assume that $budget$ is 5. The greedy algorithm initially constructs a one-node tree $T(V_T, E_T)$ containing node $t_{1,1}$ because $t_{1,1}$ has the highest profit value 9 in graph G shown in Fig. 2a. Let src be $t_{1,1}$. In iteration one, let dst be node $t_{2,3}$ because $t_{2,3}$ has the next higher profit value 7. Let $P = (t_{1,1}, v_1, v_2, v_6, t_{2,3})$ be the shortest path from src to dst in G. Because the total cost of the nodes in $\{t_{1,1}\} \cup \{t_{1,1}, v_1, v_2, v_6, t_{2,3}\}$ is 3 and is less than $budget$, P can be merged in T. Then V_T becomes $\{t_{1,1}, v_1, v_2, v_6, t_{2,3}\}$, and E_T becomes $\{(v_1, t_{1,1}), (v_1, v_2), (v_2, v_6), (v_6, t_{2,3})\}$. src is then set to $t_{2,3}$. In iteration two, let dst be node $t_{3,1}$ because $t_{3,1}$ has the next higher profit value 5. Let $P = (t_{2,3}, v_6, v_9, v_8, v_7, t_{3,1})$ be the shortest path from src to dst in G. Because the total cost of the nodes in $\{t_{1,1}, v_1, v_2, v_6, t_{2,3}\} \cup \{(t_{2,3}, v_6, v_9, v_8, v_7, t_{3,1})\}$ is 6 and is more than $budget$, P cannot be merged in T, and the algorithm terminates. Finally, $V_T = \{v_1, v_2, v_6, t_{1,1}, t_{2,3}\}$, and $E_T = \{(v_1, t_{1,1}), (v_1, v_2), (v_2, v_6), (v_6, t_{2,3})\}$.

4.2. Group-based algorithm

In each iteration of the greedy algorithm, the shortest path in G from a node in V_T to the terminal node that has the next higher profit value is considered to be added in T. However, the addition of the path may incur a higher total cost of nodes in V_T because of its longer path length. Take Fig. 3, for example. In iteration two of the greedy algorithm, $V_T = \{v_1, v_2, v_6, t_{1,1}, t_{2,3}\}$, src is $t_{2,3}$, and dst is $t_{3,1}$. When the shortest path $P = (t_{2,3}, v_6, v_9, v_8, v_7, t_{3,1})$ from src to dst in G is considered to be added in T, the total cost of nodes in V_T will increase from 3 to 6 because of the addition of v_7, v_8, and v_9. However, if we consider another path P' that has a minimum distance from the terminal node that has the next higher profit value to any node in V_T, that is, $P' = (t_{3,1}, v_7, v_5, v_2)$, the total cost of nodes in V_T will just increase from 3 to 5 because of the addition of v_5 and v_7. The tree that is merged with path P' is shown in Fig. 4. This motivates us to propose the group-based algorithm, which is extended from the greedy algorithm. In the group-based algorithm, we treat the nodes in V_T as a group. In addition, in each iteration the shortest path in G from the terminal node that has the next higher profit value to the group, that is, any node in V_T, is considered to be added into V_T. The detailed group-based algorithm is shown in Algorithm 2. The time complexity of the group-based algorithm is provided in Theorem 6.

Theorem 6. *The time complexity of the group-based algorithm is bounded in $O(m^2n^2)$, where n and m denote the numbers of nodes and terminal nodes in G, respectively.*

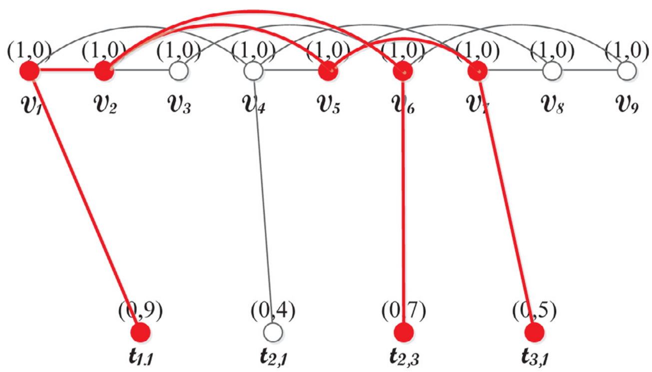


Fig. 4. A constrained node-weighted Steiner tree $T(V_T, E_T)$ constructed by the group-based algorithm in the graph G shown in Fig. 2a, where $V_T = \{v_1, v_2, v_5, v_6, v_7, t_{1,1}, t_{2,3}, t_{3,1}\}$, and $E_T = \{(v_1, v_2), (v_2, v_6), (v_2, v_5), (v_5, v_7), (v_1, t_{1,1}), (v_6, t_{2,3}), (v_7, t_{3,1})\}$.

Algorithm 2 Group-Based Algorithm ($G(V_G, E_G, w_c, w_p)$, *budget*).

Let u be the terminal node with a highest profit value in V_G
Let $T(V_T, E_T)$ be a tree with a node u
Let *tot_cost* be the cost of u
while *tot_cost* $<$ *budget* **do**
  Let W be the set of the terminal nodes $w \in V_G - V_T$ such that $w.p \geq v.p$ for all terminal nodes $v \in V_G - V_T$
  if there exists a terminal node $dst \in W$ such that $dist_G(dst, T) \leq dist_G(w, T)$ for all $w \in W$, where $dist_G(dst, T)$ (or $dist_G(w, T)$) denotes the minimum hop distance from dst (or w) to any node $x \in V_T$ in G **then**
    Let src be the node in V_T having a minimum hop distance to dst
    Let P be the shortest path from src to dst in G such that the nodes on the shortest path, excluding the end-points, are non-terminal nodes
    Let *add* be the total cost of the nodes in $S_P - V_T$, where S_P denotes the set of the nodes on P
    if *tot_cost* $+$ *add* $\leq$ *budget* **then**
      Add P into T
      tot_cost $\leftarrow$ *tot_cost* $+$ *add*
    else
      return T
    end if
  else
    return T
  end if
end while
return T

Proof. In each iteration of the group-based algorithm, at most m terminal nodes are considered to find the minimum hop distance to any node in T. Therefore, it requires at most $O(mn^2)$ to find suitable *src* and *dst* for each iteration. Because at most m iterations are in the group-based algorithm, the time complexity of the group-based algorithm is bounded in $O(m^2n^2)$. This completes the proof. □

4.3. Profit-based algorithm

In either the greedy or group-based algorithm, we always try to merge an existing tree with the terminal node that has the next higher profit value in each iteration. However, the selected terminal node may be far from the tree, that is, a longer path is required to be merged in the tree, which may incur a much higher total cost of nodes in V_T but increase little total profit. To avoid this scenario, we define a new metric for every terminal node u, called the average profit value per incremental cost of u, denoted by $u.ap$. Let T and T' be the trees

Algorithm 3 Profit-Based Algorithm ($G(V_G, E_G, w_c, w_p)$, *budget*).

Let u be the terminal node with a highest profit value in V_G
Let $T(V_T, E_T)$ be a tree with a node u
Let *tot_cost* be the cost of u
while *tot_cost* $<$ *budget* **do**
  Evaluate every terminal node u's $u.ap$
  Let W be the set of the terminal nodes $w \in V_G - V_T$ such that $w.ap \geq v.ap$ for all terminal nodes $v \in V_G - V_T$
  if there exists a terminal node $dst \in W$ such that $dist_G(dst, T) \leq dist_G(w, T)$ for all $w \in W$, where $dist_G(dst, T)$ (or $dist_G(w, T)$) denotes the minimum hop distance from dst (or w) to any node $x \in V_T$ in G **then**
    Let src be the node in V_T having a minimum hop distance to dst
    Let P be the shortest path from src to dst in G such that the nodes on the shortest path, excluding the end-points, are non-terminal nodes
    Let *add* be the total cost of the nodes in $S_P - V_T$, where S_P denotes the set of the nodes on P
    if *tot_cost* $+$ *add* $\leq$ *budget* **then**
      Add P into T
      tot_cost $\leftarrow$ *tot_cost* $+$ *add*
    else
      return T
    end if
  else
    return T
  end if
end while
return T

before and after the terminal node u is merged by the shortest path in G from a node in T to u. The $u.ap$ is defined as follows:

$$u.ap = \frac{profit(T')}{cost(T') - cost(T)}, \quad (1)$$

where *profit*(T') denotes the total profit of nodes in T', and *cost*(T) (or *cost*(T')) denotes the total cost of nodes in T (or T'). By the metric, we therefore propose the profit-based algorithm, which is extended from the group-based algorithm. In each iteration of the profit-based algorithm, we evaluate every terminal node u's $u.ap$. Then the shortest path in G from the terminal node, which has the next higher average profit value per incremental cost, to any node in the existing tree $T(V_T, E_T)$, is considered to be added into T. The detailed profit-based algorithm is shown in Algorithm 3. The time complexity of the profit-based algorithm is provided in Theorem 7.

Theorem 7. *The time complexity of the profit-based algorithm is bounded in $O(m^2n^2)$, where n and m denote the numbers of nodes and terminal nodes in G, respectively.*

Proof. In each iteration of the profit-based algorithm, except for the evaluation of the average profit value, the other part requires $O(mn^2)$ due to the similarity of the group-based algorithm. In addition, to evaluate every node u's $u.ap$, the shortest path from u to an existing tree in G is required to be found and evaluated. Because there are at most m terminal nodes, it requires $O(mn^2)$ for the evaluation. We thus have that it requires $O(mn^2) + O(mn^2) = O(mn^2)$ for each iteration. Because at most m iterations are in the profit-based algorithm, the time complexity of the profit-based algorithm is bounded in $O(m^2n^2)$. This completes the proof. □

Take Fig. 5, for example. Assume that *budget* is 5. Initially, a one-node tree $T(V_T, E_T)$ containing node $t_{1,1}$ is constructed. In iteration one, we have that $t_{2,1}.ap = 6.5$, $t_{2,3}.ap = 5.33$, and $t_{3,1}.ap = 4.67$. To evaluate $t_{2,1}.ap$, the *cost*(T) and *cost*(T') must be evaluated first,

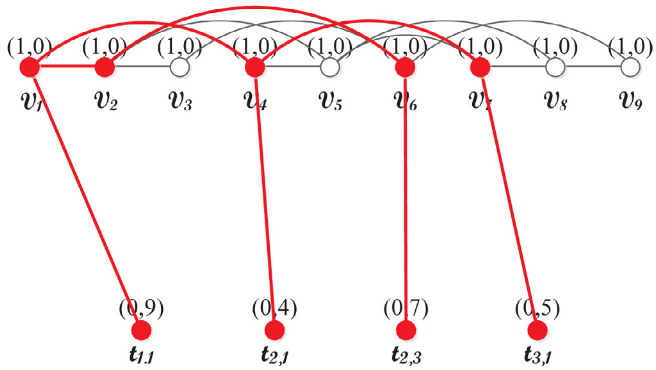


Fig. 5. A constrained node-weighted Steiner tree $T(V_T, E_T)$ constructed by the profit-based algorithm in the graph G shown in Fig. 2a, where $V_T = \{v_1, v_2, v_4, v_6, v_7, t_{1,1}, t_{2,1}, t_{2,3}, t_{3,1}\}$, and $E_T = \{(v_1, v_2), (v_1, v_4), (v_2, v_6), (v_4, v_7), (v_1, t_{1,1}), (v_4, t_{2,1}), (v_6, t_{2,3}), (v_7, t_{3,1})\}$.

where T' is the tree T after merging the terminal node $t_{2,1}$ by the shortest path in G. Clearly, the shortest path from $t_{2,1}$ to T is ($t_{2,1}$, v_4, v_1, $t_{1,1}$). Therefore, *cost*(T) is 0 and *cost*(T') is 2 because v_1 and v_4 are non-terminal nodes. In addition, *profit*(T') is $9+4=13$ because $t_{2,1}$ is included in T'. Therefore, $t_{2,1}.ap = \frac{13}{2-0} = 6.5$. The $t_{2,3}.ap$ and $t_{3,1}.ap$ can be evaluated in the same way. Because $t_{2,1}.ap$ is greater than $t_{2,3}.ap$ and $t_{3,1}.ap$, we merge T with $t_{2,1}$. Therefore, *dst* and *src* are set to $t_{2,1}$ and $t_{1,1}$, respectively. Then the edges not in T but on the shortest path in G from $t_{2,1}$ to $t_{1,1}$, including (v_4, $t_{2,1}$), (v_1, v_4), and (v_1, $t_{1,1}$), are inserted into T. Then we have that $V_T = \{t_{1,1}, v_1, v_4, t_{2,1}\}$, $E_T = \{(v_1, t_{1,1}), (v_1, v_4), (v_4, t_{2,1})\}$, and $tot\_cost = 2$. Following the same process, in iteration two, we insert the edges on the shortest path in G from $t_{3,1}$ to v_4 because $t_{3,1}.ap = 18$ and $t_{2,3}.ap = 10$. In iteration three, we insert the edges on the shortest path in G from $t_{2,3}$ to v_1 because $tot\_cost = 5$ is not greater than *budget*. Finally, the tree is constructed as shown in Fig. 5.

5. Performance evaluation

Because the simulators, including avrora, castalia, tossim, and cooja [35–37], are often used to simulate physical, MAC, and network layer protocols for an existing wireless sensor networks, in this paper, we thus develop another simulator implemented by C language to evaluate the performance of the proposed methods for constructing new wireless sensor networks. In the simulation, the sensing field was divided into 20×20 grids with side length 1. The critical grids were randomly selected from the grids in the sensing field. The weight on each critical grid was randomly chosen from the interval [1, 100]. In addition, 800 points that were allowed to deploy sensors were randomly chosen within the sensing field. There were at most n sensors that could be deployed in the sensing field, where each sensor had sensing range R_s and transmission range R_t. In the simulation, we evaluated the performance with our proposed methods, including Reduction+GA, Reduction+GBA, and Reduction+PBA, where the Reduction+GA, the Reduction+GBA, and the Reduction+PBA denoted the proposed reductions by applying the greedy algorithm, the group-based algorithm, and the profit-based algorithm, respectively. The total weight of the critical grids covered by deployed sensors was evaluated with the proposed methods in Sections 5.1–5.4. In addition, the total number of deployed sensors was evaluated with the proposed methods in Section 5.5, when n was set large enough such that all critical grids can be covered. Finally, we have provided a simulation to compare our proposed methods with an exhaustive search [38,39] in Section 5.6. Although the exhaustive search can find an optimal solution for the weighted-critical-square-grid coverage problem, it requires exponential time. The following empirical data were obtained by averaging the data of 100 sensing fields. The confidence interval for the 95% confidence level of the total weight of the critical grids covered by deployed sensors (or the total number of deployed sensors) is within 5% of the mean value.

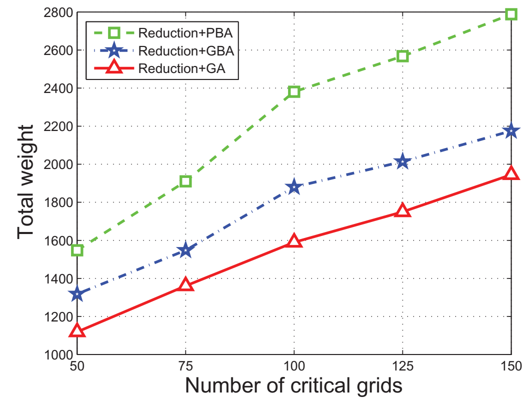


Fig. 6. Comparison of the instances that have $R_s = \sqrt{2}$, $R_t = 1$, $n = 75$, and the number of critical grids ranges from 50 to 150.

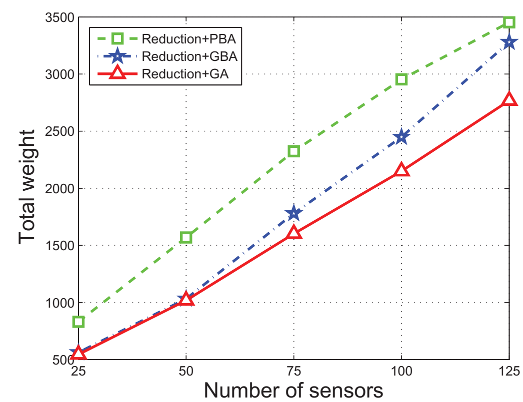


Fig. 7. Comparison of the instances that have $R_s = \sqrt{2}$, $R_t = 1$, the number of critical grids is equal to 100, and n ranges from 25 to 125.

5.1. Number of critical grids

Fig. 6 illustrates the results in the sensing fields that have the number of critical grids ranging from 50 to 150 when $R_s = \sqrt{2}$, $R_t = 1$, and $n = 75$. The Reduction+PBA provides a better performance than the Reduction+GA and the Reduction+GBA. This is because the PBA considers using minimum sensors to achieve great total profit for solving the constrained node-weighted Steiner tree problem. Thus, the maximum total profit is achieved by the Reduction+PBA while a fixed number of sensors are used. In addition, the Reduction+GA provides the worst performance among the proposed methods. This stems from the fact that in each iteration of the GA, the addition of a path to a tree may cause higher total cost. Therefore, more sensors are required to increase total profit in the Reduction+GA. In addition, the higher the number of critical grids, the higher the total weight of the proposed methods. This is because it has more probability of covering critical grids by the deployed sensors.

5.2. Number of sensors

Fig. 7 shows the results when $R_s = \sqrt{2}$, R_t= 1, the number of critical grids is equal to 100, and n ranges from 25 to 125. The Reduction+PBA and the Reduction+GA provide the best and the worst performances among the proposed methods. In addition, the higher the number of sensors, the higher the total weight of the proposed methods. This is because more sensors can be used to cover critical grids.

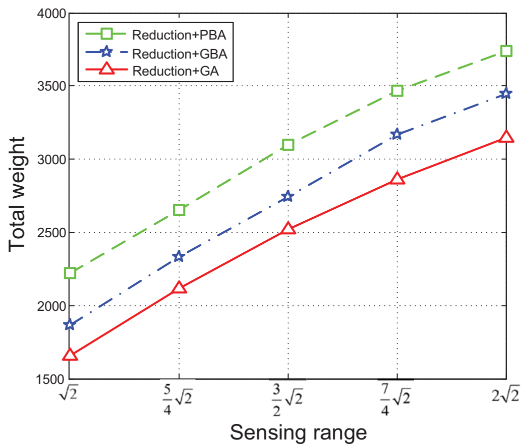


Fig. 8. Comparison of the instances having $R_t = 1$, n =75, the number of critical grids is equal to 100, and R_s ranges from $\sqrt{2}$ to $2\sqrt{2}$.

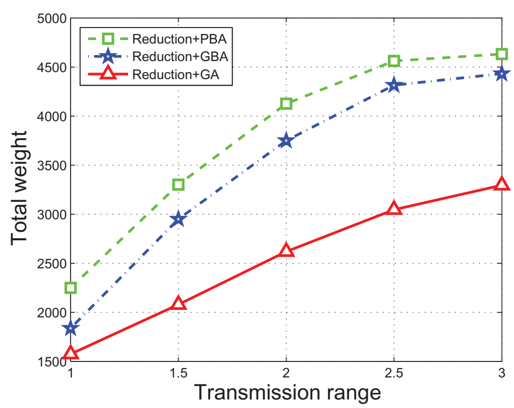


Fig. 9. Comparison of the instances that have $R_s = \sqrt{2}$, $n = 75$, the number of critical grids is equal to 100, and R_t ranges from 1 to 3.

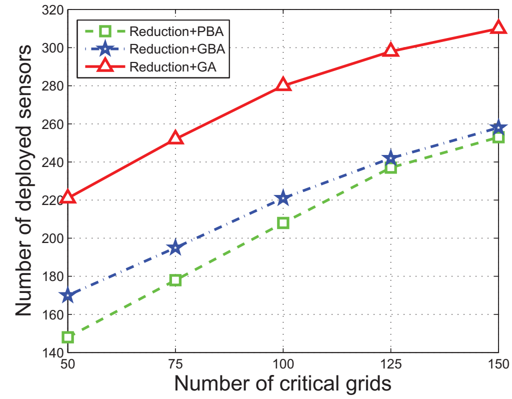


Fig. 10. Comparison of the instances that have $R_s = \sqrt{2}$, $R_t = 1$, and the number of critical grids ranging from 50 to 150 when n is set large enough such that all critical grids can be covered.

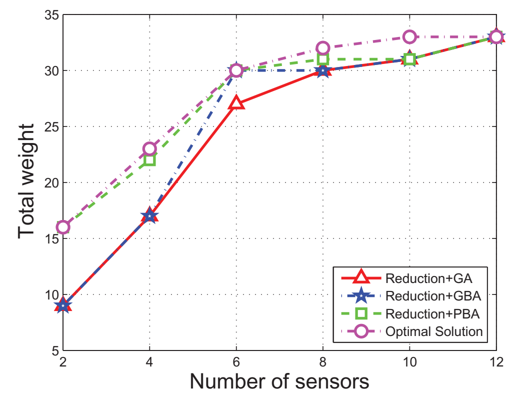


Fig. 11. Comparison of the instances that have a sensing field divided into 3 × 3 grids with side length $F1$, $R_s = \sqrt{2}$, $R_t = 1$, the number of critical grids is equal to 5, and n ranges from 2 to 12.

5.3. Sensing range

Fig. 8 illustrates the results in the sensing fields that have R_s ranging from $\sqrt{2}$ to $2\sqrt{2}$ when R_t =1, n = 75, and the number of critical grids is equal to 100. Clearly, the Reduction+PBA provides the best performance among the proposed methods. In addition, the larger the sensing range of each sensor, the higher the total weight that can be achieved by the proposed methods. This is because when the sensing range increases, the deployed sensors have more probability of covering more critical grids.

5.4. Transmission range

Fig. 9 shows the results in the sensing fields that have R_t ranging from 1 to 3 when $R_s = \sqrt{2}$, n =75, and the number of critical grids is equal to 100. The Reduction+PBA has the best performance. In addition, the larger the transmission range, the higher the total weight achieved by the proposed methods. This is because fewer sensors are required to keep the sensors connected, and thus, more sensors can be used to cover critical grids.

5.5. Number of deployed sensors versus number of critical grids

Fig. 10 illustrates the results in the sensing fields that have the number of critical grids ranging from 50 to 150 when $R_s = \sqrt{2}$, $R_t =$ 1, and n is set large enough such that all critical grids can be covered. The Reduction+PBA requires fewer sensors to cover all critical grids than the Reduction+GA and the Reduction+GBA. This is because the Reduction+PBA considers using fewer sensors to cover critical grids and form a connected wireless sensor network, as discussed before. In addition, as we expected, the more critical grids, the more sensors are required to be deployed in our proposed methods.

5.6. Optimal solution versus proposed methods

Fig. 11 shows the results in the sensing fields, each divided into 3 × 3 grids with side length 1, when $R_s = \sqrt{2}$, $R_t = 1$, the number of critical grids is equal to 5, and n ranges from 2 to 12. The weight on each critical grid was randomly chosen from the interval [1, 10]. In addition, 18 points that were allowed to deploy sensors were randomly chosen within the sensing field. In Fig. 11, the Reduction+PBA provides better performance than the Reduction+GBA and the Reduction+GA, as observed in Fig. 7. In addition, the result of the Reduction+PBA is close to that of the optimal solution.

6. Conclusion

In the paper, we investigated the weighted-critical-square-grid coverage problem, which is the problem of using limited sensors to construct a wireless sensor network such that the total weight of the covered critical square grids is maximized. The problem was shown to be NP-complete. In addition, a reduction that transforms the weighted-critical-square-grid coverage problem into the constrained node-weighted Steiner tree problem was proposed. Once a solution to the constrained node-weighted Steiner tree problem is obtained, the solution can be used to select the points that are allowed to deploy sensors for the weighted-critical-square-grid coverage problem. We also showed that the constrained node-weighted

Steiner tree problem is NP-complete. In addition, the greedy algorithm (GA), the group-based algorithm (GBA), and the profit-based algorithm (PBA), were proposed for the constrained node-weighted Steiner tree problem.

In the simulation, we evaluated the performance with our proposed methods, including Reduction+GA, Reduction+GBA, and Reduction+PBA, in terms of the total weight of the critical grids covered by deployed sensors, where the Reduction+GA, the Reduction+GBA, and the Reduction+PBA denoted the proposed reductions by applying the greedy algorithm, the group-based algorithm, and the profit-based algorithm, respectively. The simulation results showed that the Reduction+PBA had a higher total weight than the others. We also evaluated the performance with the proposed methods in terms of the number of deployed sensors if the number of sensors was large enough to cover all critical grids. The simulation results showed that the Reduction+PBA used fewer sensors than the others for deployment.

Acknowledgments

This work was supported by the Ministry of Science and Technology under Grant MOST 103-2221-E-151-002.

References

[1] A.C. Valera, W.-S. Soh, H.-P. Tan, Survey on wakeup scheduling for environmentally-powered wireless sensor networks, Comput. Commun. 52 (0) (2014) 21–36.
[2] Y.K. An, S.-M. Yoo, C. An, B.E. Wells, Rule-based multiple-target tracking in acoustic wireless sensor networks, Comput. Commun. 51 (0) (2014) 81–94.
[3] G. Huang, A.Y. Zomaya, F.C. Delicato, P.F. Pires, Long term and large scale time synchronization in wireless sensor networks, Comput. Commun. 37 (0) (2014) 77–91.
[4] F. Bouabdallah, N. Bouabdallah, R. Boutaba, Reliable and energy efficient cooperative detection in wireless sensor networks, Comput. Commun. 36 (5) (2013) 520–532.
[5] F.M. Al-Turjman, H.S. Hassanein, M.A. Ibnkahla, Efficient deployment of wireless sensor networks targeting environment monitoring applications, Comput. Commun. 36 (2) (2013) 135–148.
[6] Z. Sheng, C. Mahapatra, C. Zhu, V. Leung, Recent advances in industrial wireless sensor networks toward efficient management in IOT, IEEE Access 3 (2015) 622–637.
[7] S. Kumar, T.H. Lai, A. Arora, Barrier coverage with wireless sensors, in: Proceedings of the 11th Annual International Conference on Mobile Computing and Networking, New York, NY, USA, 2005, pp. 284–298.
[8] Y.-L. Lai, J.-R. Jiang, Barrier coverage with optimized quality for wireless sensor networks, in: Proceedings of Wireless Personal Multimedia Communications, 2012, pp. 192–196.
[9] C. Shen, W. Cheng, X. Liao, S. Peng, Barrier coverage with mobile sensors, in: Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks, 2008, pp. 99–104.
[10] S. Silvestri, Mobibar: barrier coverage with mobile sensors, in: Proceedings of IEEE Global Telecommunications Conference, 2011, pp. 1–6.
[11] A. Saipulla, C. Westphal, B. Liu, J. Wang, Barrier coverage of line-based deployed wireless sensor networks, in: Proceedings of IEEE INFOCOM, 2009, pp. 127–135.
[12] X. Bai, Z. Yun, D. Xuan, T. Lai, W. Jia, Deploying four-connectivity and full-coverage wireless sensor networks, in: Proceedings of IEEE INFOCOM, 2008.
[13] Q. Yang, S. He, J. Li, J. Chen, Y. Sun, Energy-efficient probabilistic full coverage in wireless sensor networks, in: Proceedings of IEEE Global Communications Conference, 2012, pp. 591–596.
[14] T. Yan, Y. Gu, T. He, J.A. Stankovic, Design and optimization of distributed sensing coverage in wireless sensor networks, J. ACM Trans. Embed. Comput. Syst. 7 (3) (2008) 33:1–33:40.
[15] X. Bai, S. Kumar, D. Xuan, Z. Yun, T.H. Lai, Deploying wireless sensors to achieve both coverage and connectivity, in: Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing, New York, NY, USA, 2006, pp. 131–142.
[16] X. Bai, D. Xuan, Z. Yun, T.H. Lai, W. Jia, Complete optimal deployment patterns for full-coverage and k-connectivity wireless sensor networks, in: Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2008, pp. 401–410.
[17] C.-Y. Chang, Y.-C. Chen, H.-R. Chang, Obstacle-resistant deployment algorithms for wireless sensor networks, IEEE Trans. Veh. Technol. 58 (6) (2009) 2925–2941.
[18] K. Chakrabarty, S. Member, S.S. Iyengar, E. Cho, Grid coverage for surveillance and target location in distributed sensor networks, IEEE Trans. Comput. 51 (2002) 1448–1453.
[19] Y.-C. Wang, C.-C. Hu, Y.-C. Tseng, Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks, in: Proceedings of the First International Conference on Wireless Internet, 2005, pp. 114–121.
[20] W.-C. Ke, B.-H. Liu, M.-J. Tsai, The critical-square-grid coverage problem in wireless sensor networks is NP-complete, J. Comput. Netw. 55 (9) (2011a) 2209–2220.
[21] W.-C. Ke, B.-H. Liu, M.-J. Tsai, Efficient algorithm for constructing minimum size wireless sensor networks to fully cover critical square grids, IEEE Trans. Wireless Commun. 10 (4) (2011b) 1154–1164.
[22] B.-H. Liu, K.-W. Su, Enhanced algorithms for deploying the minimum sensors to construct a wireless sensor network having full coverage of critical square grids, ACM/Springer Wireless Netw. 20 (2) (2014) 331–343.
[23] Greenorbs, 2011, [Online]. Available: http://www.greenorbs.org (accessed 17.10.14).
[24] N. Riaz, M. Ghavami, Analytical performance evaluation of ultra-wideband multiple access schemes for different wireless sensor network application environments, IEEE Trans. Microw. Theory Tech. 3 (9) (2009) 1473–1487.
[25] A. Farooqi, F. Khan, Intrusion detection systems for wireless sensor networks: a survey, in: Communication and Networking, in: Communications in Computer and Information Science, vol. 56, Springer, Berlin Heidelberg, 2009, pp. 234–241.
[26] A. Diwadkar, U. Vaidya, Performance limitation in multi-agent estimation with limited sensor measurements, in: Proceedings of IEEE Conference on Decision and Control, 2010, pp. 6614–6619.
[27] A. Muis, R. Prasetya, Implementation of landmark-based localization on wall-based environment for mobile robot with limited sensor resources, in: Proceedings of IEEE TENCON, 2011, pp. 1050–1054.
[28] D. He, C. Chen, S. Chan, J. Bu, SDRP: a secure and distributed reprogramming protocol for wireless sensor networks, IEEE Trans. Ind. Electron. 59 (11) (2012) 4155–4163.
[29] J.S. McLACHLAN, J.J. HELLMANN, M.W. SCHWARTZ, A framework for debate of assisted migration in an era of climate change, Conservation Biology 21 (2) (2007) 297–302.
[30] N. Heo, P.K. Varshney, Energy-efficient deployment of intelligent mobile sensor networks, IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans 35 (1) (2005) 78–92.
[31] B.N. Clark, C.J. Colbourn, D.S. Johnson, Unit disk graphs, Discret. Math. 86 (1–3) (1991) 165–177.
[32] C.-Y. Chang, C.-Y. Hsiao, C.-Y. Chang, The k-barrier coverage mechanism in wireless visual sensor networks, in: Proceedings of IEEE Wireless Communications and Networking Conference, 2012, pp. 2318–2322.
[33] A. Moss, Y. Rabani, Approximation algorithms for constrained node weighted steiner tree problems, in: Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, 2001, pp. 373–382.
[34] M. Thorup, Undirected single-source shortest paths with positive integer weights in linear time, J. ACM 46 (3) (1999) 362–394.
[35] Avrora, [Online]. Available: http://compilers.cs.ucla.edu/avrora/ (accessed 17.12.08).
[36] Castalia, [Online]. Available: https://castalia.forge.nicta.com.au (accessed 30.03.11).
[37] Cooja, [Online]. Available: http://www.contiki-os.org/start.html (accessed 15.11.13).
[38] R. Kulkarni, G. Venayagamoorthy, Bio-inspired algorithms for autonomous deployment and localization of sensor nodes, IEEE Trans. Syst. Man Cybern C: Appl. Rev. 40 (6) (2010) 663–675.
[39] D. Bajovic, B. Sinopoli, J. Xavier, Sensor selection for event detection in wireless sensor networks, IEEE Trans. Signal Process. 59 (10) (2011) 4938–4953.