Theoretical Computer Science 410 (2009) 812–817

Contents lists available at ScienceDirect

Theoretical Computer Science

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

Algorithms for connected set cover problem and fault-tolerant connected set cover problem

Zhao Zhang [a,*], Xiaofeng Gao [b], Weili Wu [b]

[a] College of Mathematics and System Sciences, Xinjiang University, Urumqi, Xinjiang, 830046, PR China

[b] Department of Computer Science, University of Texas at Dallas, Richardson, TX, 75080, USA

ARTICLE INFO

Article history:
Received 12 September 2008
Accepted 5 November 2008
Communicated by D.-Z. Du

Keywords:
Reserve system
Connected set cover
Fault-tolerant

ABSTRACT

Given a set V of elements, $\mathcal{S}$ a family of subsets of V, and G a connected graph on vertex set $\mathcal{S}$,a connected set cover (CSC) is a subfamily $\mathcal{R}$ of $\mathcal{S}$ such that every element in V is covered by at least one set of $\mathcal{R}$, and the subgraph $G[\mathcal{R}]$ of G induced by $\mathcal{R}$ is connected. If furthermore $G[\mathcal{R}]$ is k-connected and every element in V is covered by at least m sets in $\mathcal{R}$, then $\mathcal{R}$ is a (k, m)-CSC. In this paper, we present two approximation algorithms for the minimum CSC problem, and one approximation algorithm for the minimum $(2, m)$-CSC problem. Performance ratios are analyzed. These are the first approximation algorithms for CSC problems in general graphs with guaranteed performance ratios.

1. Introduction

This paper studies approximation algorithms for minimum connected set cover problems (MCSC).

Let V be a set of elements, and $\mathcal{S}$ be a family of subsets of V such that $\bigcup_{S\in\mathcal{S}} S = V$. A *set cover (SC) with respect to* $(V, \mathcal{S})$ is a sub-family $\mathcal{R}$ of $\mathcal{S}$ such that every element $v \in V$ is in some set $S \in \mathcal{R}$. We say that S covers v. Let G be a connected graph on vertex set $\mathcal{S}$. A *connected set cover with respect to* $(V, \mathcal{S}, G)$ (abbreviated as CSC) is a set cover $\mathcal{R}$ with respect to $(V, \mathcal{S})$ such that the subgraph of G induced by $\mathcal{R}$ is connected. We use the terminology 'set' and 'vertex' interchangeably when talking about elements in $\mathcal{S}$.

The Minimum Set Cover problem (MSC) has a lot of applications in the real world. For example, in establishing a biodiversity reserve system, a set of reserves (or protected areas) are chosen from candidate sites such that all species are represented at the reserves. For economical reason, the object is to minimize the number of chosen reserves. This problem can be modeled as an MSC: the set of all species is V; for each candidate site, the set of species contained in it is a subset in $\mathcal{S}$; establishing an economic reserve system is equivalent to finding a minimum set cover.

However, the above model is not sufficient for long-term persistence of species. In fact, without more constraints, the reserve system found by solving MSC is almost always highly fragmented (that is, the system might have many disconnected sites), and thus is more vulnerable to natural and biological invasions [19]. To solve this problem, corridors are established which facilitate dispersal and colonization between reserves. Experimental studies show that the presence of corridors increases species richness [6]. In view of this consideration, many researchers incorporated connectivity criteria into their objective functions. The connectivity criteria used in these works include incorporating a connectivity function in a multi-objective function [21], minimizing distances between pairs of sites [16,17], minimizing the sum of all pairwise distances between sites [2,15,17], minimizing the boundary length of a network [18], and minimizing some function which is the combination of the boundary length and the area [14,19]. Methods used to solve these models include greedy strategy,

* Corresponding author. Tel.: +86 13669969821.
E-mail address: hxhzz@163.com (Z. Zhang).

0304-3975/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2008.11.005

integer programming, simulated annealing, and some combination of them. However, none of them has a performance guarantee.

Minimum Connected Set Cover problem (MCSC) can serve as a simple model for the reserve system problem with connectivity constraint: the potential corridors connect $\mathcal{S}$ into a graph G. Finding an economic reserve system with connectivity constraint is equivalent to finding a minimum connected set cover.

It is well known that the MSC problem is NP-hard [9], and can not be approximated within a factor of $(1-\varepsilon)\ln n$ for any $\varepsilon > 0$ unless $NP \subseteq DTIME(n^{\log\log n})$ [8], where $n = |V|$. Since MSC is a special case of MCSC (taking G to be a complete graph), MCSC is also NP-hard and is not $(1-\varepsilon)\ln n$-approximable. Furthermore, Shuai and Hu [20] showed that even when at most one vertex of the graph G has degree greater than two, the MCSC problem is still non-$(1-\varepsilon)\ln n$-approximable. In the case that the graph is a path, Shuai and Hu gave two polynomial-time algorithms. In the case that the graph has exactly one vertex of degree greater than two, they proposed a $(1+\ln n)$-approximation algorithm. For the general case, there is no known approximation algorithm with guaranteed performance ratio. A theoretical study on CSC was done by Cerdeira and Pinto [3], who studied some valid inequalities for the convex hull of the set of incidence vectors of CSC.

This paper gives the first approximation algorithms for the MCSC problem in a general graph, implementing a new parameter D_c. For any two sets $S_1, S_2 \in \mathcal{S}$, $dist_G(S_1, S_2)$ is the length of a minimum (S_1, S_2)-path in G, where length refers to the number of edges on this path. Two sets $S_1, S_2 \in \mathcal{S}$ are said to be *cover-adjacent* if $S_1 \cap S_2 \neq \emptyset$. Let $D_c(G) = \max\{dist_G(S_1, S_2) \mid S_1, S_2 \in \mathcal{S} \text{ and } S_1, S_2 \text{ are cover-adjacent}\}$. In this paper, we present two approximation algorithms for the MCSC problem. One is a two-step algorithm. It first finds an SC using an α-approximation algorithm, and then connects them with a Steiner Minimum Tree with Minimum Number of Steiner Points (SMT-MSP) using a β-approximation algorithm. The performance ratio of this algorithm is $\alpha+\beta+\alpha\beta(D_c-1)$. The second algorithm uses a greedy strategy, and the performance ratio is $1 + D_c(G) \cdot H(\gamma-1)$, where H is the harmonic function, and $\gamma = \max\{|S| \mid S \in \mathcal{S}\}$. In many cases, $D_c = 1$. For example, if two reserves containing a same species are regarded to be adjacent, then $D_c = 1$. In such cases, the two algorithms given in this paper has performance ratio $\alpha+\beta$ and $1 + H(\gamma-1)$ respectively.

Then, we consider the fault-tolerant CSC problem. For a CSC $\mathcal{R}$, if the subgraph of G induced by $\mathcal{R}$ is k-connected, and every element of V is covered by at least m sets of $\mathcal{R}$, then $\mathcal{R}$ is a (k, m)-*connected set cover* ((k, m)-CSC for short). If a reserve system takes the form of a (k, m)-CSC, then every species is represented in at least m reserves, and the connection among the reserves is more fault tolerant in the face of disasters.

In this paper, we present a greedy algorithm for the minimum $(2, m)$-CSC problem, using a parameter $PD(G)$. Given three vertices u, v, w in a graph G, define the *pair distance between* u *and* $\{v, w\}$, denoted by $dist(u; v, w)$, to be the shortest length of a pair of disjoint (u, v)-path and (u, w)-path. In another words, it is the length of the shortest (v, w)-path through vertex u. The *pair diameter* of a graph G is $PD(G) = \min\{dist(u; v, w) \mid u, v, w \text{ are three distinct vertices in } V(G)\}$. Our algorithm has performance ratio $(PD(G)-1)(1 + H(\gamma-1))$.

The remainder of the paper is organized as follows. In Section 2, we present the two-step algorithm for MCSC. In Section 3, we present the greedy algorithm for MCSC. In Section 4, we present the greedy algorithm for minimum (k, m)-CSC. Performance ratios are analyzed in the corresponding sections. In the last section, we make a discussion and propose some future work.

2. A two-step algorithm for MCSC

The idea of the two-step algorithm is simple. It combines an algorithm finding MSC with an algorithm computing SMT-MSP.

The MSC problem has long been a research topic in combinatorial optimization, and there are a lot of approximation algorithms for it. In some special cases, a better performance ratio than $\ln n$ can be obtained. See for example [1] page 424–425.

In an SMT-MSP problem, we are given a set of points called *terminals*. The objective is to find a Steiner tree spanning all the terminals such that the number of Steiner points is minimum. For a metric space, the problem of SMT-MSP with bounded edge-lengths were studied in [4,7,12,13]. The best known approximation ratio for this problem is 3 [7], and a randomized algorithm was also presented in [7] with a performance ratio 2.5 at probability at least 1/2. The SMT-MSP problem in a general graph is a special case of the Node Weighted Steiner Tree problem (NWST), in which every node has a weight, the objective is to choose some Steiner nodes to connect all the terminals such that the total weight of the Steiner nodes is a minimum. Clearly, an NWST problem with all nodes having weight one is the SMT-MSP problem. Klein and Ravi [11] showed that the NWST problem cannot be approximated to within less than a logarithmic factor unless $NP \subseteq DTIME(k^{\log\log k})$, and presented a $2\ln k$-approximation algorithm by inventing an original idea of *spider decomposition*, where k is the number of terminals. Later, Guha and Khuller [10] generalized this idea to *branch spiders*, and gave a $(1.35+\varepsilon)\ln k$-approximation algorithm, which is the best guarantee known up to now. In some special case, the performance ratio can be better. For example, in a unit disk graph, a 3.875-approximation algorithm was known [23].

The two-step algorithm is depicted in the following.

Next, we analyze the performance ratio of the above algorithm.

Theorem 1. *Suppose the approximation ratio of $\mathcal{A}$ and $\mathcal{B}$ are α and β respectively. Then the performance ratio for Algorithm 1 is $\alpha+\beta+\alpha\beta(D_c-1)$.*

Algorithm 1

Input: $(V, \mathcal{S}, G)$; an algorithm $\mathcal{A}$ computing a minimum set cover; an algorithm $\mathcal{B}$ computing a Steiner tree with minimum number of Steiner points.
Output: A connected set cover $\mathcal{R}$.
1: Use $\mathcal{A}$ to compute a set cover $\mathcal{R}_1$ with respect to $(V, \mathcal{S})$.
2: Use $\mathcal{B}$ to compute a Steiner tree T in G with terminal set $\mathcal{R}_1$. Let $\mathcal{R}_2$ be the Steiner points of T.
3: Output $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$.

Proof. Let $\mathcal{R}^*$ be an optimal solution to MCSC, and $\mathcal{R}_2^*$ be a Steiner tree of G connecting terminal set $\mathcal{R}_1$ with a minimum number of Steiner points. Since $\mathcal{R}^*$ is also a set cover with respect to $(V, \mathcal{S})$, we have

$$|\mathcal{R}_1| \le \alpha |\mathcal{R}^*|. \tag{1}$$

Let S be a set in $\mathcal{R}_1$. Suppose v is an element of V covered by S, and S^* is a set in $\mathcal{R}^*$ covering v. Then S, S^* are cover-adjacent, and thus $dist_G(S, S^*) \le D_c$. By adding at most $D_c - 1$ vertices of G connects S to S^*. It follows that by adding at most $(D_c - 1)|\mathcal{R}_1|$ vertices, all vertices in $\mathcal{R}_1$ are connected to $\mathcal{R}^*$. Since $G[\mathcal{R}^*]$ is connected, we have

$$|\mathcal{R}_2^*| \le |\mathcal{R}^*| + (D_c - 1)|\mathcal{R}_1|. \tag{2}$$

Combining inequalities (1) and (2) with $|\mathcal{R}_2| \le \beta |\mathcal{R}_2^*|$, the approximation ratio follows. □

3. A greedy algorithm for MCSC

In this section, we present a greedy algorithm for MCSC depicted in Algorithm 2. In this algorithm, $\mathcal{R}$ records the sets which have been chosen and U records the set of elements of V which have been covered. For $\mathcal{R} \ne \emptyset$ and a set $S \in \mathcal{S} \setminus \mathcal{R}$, an $\mathcal{R}$-S path is a path in G such that its initial vertex is in $\mathcal{R}$, its end vertex is S, and all the other vertices on this path are in $\mathcal{S} \setminus \mathcal{R}$. Clearly, for a shortest $\mathcal{R}$-S path P_S, it has exactly $|P_S|$ vertices in $\mathcal{S} \setminus \mathcal{R}$, where $|P_S|$ is the number of edges in P_S. We use $C(P_S)$ to denote the set of elements of $V \setminus U$ which are covered by vertices on P_S. Define

$$e(P_S) = \frac{|P_S|}{|C(P_S)|}.$$

Algorithm 2

Input: $(V, \mathcal{S}, G)$.
Output: A connected set cover $\mathcal{R}$.
1: Choose $S_0 \in \mathcal{S}$ such that $|S_0|$ is maximum. $\mathcal{R} = \{S_0\}$, $U = S_0$.
2: **while** $V \setminus U \ne \emptyset$ **do**
3: For each $S \in \mathcal{S} \setminus \mathcal{R}$ which is cover-adjacent with a set in $\mathcal{R}$, compute a shortest $\mathcal{R}$-S path P_S. Choose S such that $e(P_S)$ is minimum. Add all sets on P_S into $\mathcal{R}$, $U = U \cup C(P_S)$.
4: **end while**
5: Output $\mathcal{R}$.

The output $\mathcal{R}$ of Algorithm 2 is obviously a connected set cover for $(V, \mathcal{S}, G)$. Next, we analyze the performance ratio.

Theorem 2. *Algorithm 2 has performance ratio $1 + D_c(G) \cdot H(\gamma - 1)$, where $\gamma = \max\{|S| \mid S \in \mathcal{S}\}$, and H is the harmonic function.*

Proof. Suppose S_i is the set chosen in the ith iteration (S_0 is the initial set chosen in line 1). Let $\mathcal{S}_i$ be the set of sets added to $\mathcal{R}$ in the ith iteration (that is, the vertices on P_{S_i} which is not already in $\mathcal{R}$). Then $\mathcal{R}_k = \bigcup_{i=0}^{k} \mathcal{S}_i$ is the set of sets chosen after the kth iteration. Suppose Algorithm 2 runs K rounds. Then $\mathcal{R}_K$ is the output of the algorithm. When S_i is chosen, we assign each element $v \in C(P_{S_i})$ a weight $w(v) = e(P_{S_i})$ for $i \ge 1$ and $w(v) = 1/|S_0|$ for $i = 0$. Then each element $v \in V$ is assigned a weight exactly once, and

$$\sum_{v \in V} w(v) = \sum_{i=0}^{K} \sum_{v \in C(P_{S_i})} w(v) = \sum_{i=0}^{K} \sum_{v \in C(P_{S_i})} \frac{|P_{S_i}|}{|C(P_{S_i})|} = \sum_{i=0}^{K} |P_{S_i}| = |\mathcal{R}_K|. \tag{3}$$

(Here, P_{S_0} is the path with only one vertex S_0, $C(P_{S_0}) = S_0$, and thus $|P_{S_0}|/|C(P_{S_0})| = 1/|S_0|$ is exactly the weight assigned to elements covered by S_0).

Suppose $\mathcal{R}^* = \{S_1^*, \dots, S_{opt}^*\}$ is an optimal solution to the CSC problem. Set $N_1 = S_1^*$, and for $i = 2, \dots, opt$, set $N_i = S_i^* \setminus (\bigcup_{j=1}^{i-1} N_j)$. Clearly, $N_i \cap N_j = \emptyset$ for $i \ne j$. Since $\mathcal{R}^*$ covers all elements of V, we see that $N_1, \dots, N_{opt}$ is a partition of V. It follows that

$$\sum_{v \in V} w(v) = \sum_{k=1}^{opt} \sum_{v \in N_k} w(v). \tag{4}$$

Next, we show that for each $k \in \{1, \ldots, opt\}$,

$$\sum_{v \in N_k} w(v) \leq 1 + D_c(G) \cdot H(\gamma - 1). \tag{5}$$

Let $n_0 = |N_k|$, and for $i = 1, \ldots, opt$ let n_i be the number of elements in N_k which are not covered after the ith iteration. For $i = 1, \ldots, opt$, after the ith iteration, $n_{i-1} - n_i$ elements of N_k are covered and each such an element is assigned a weight

$$e(P_{S_i}) \leq e(P_{S_k^*}) = \frac{|P_{S_k^*}|}{|C(P_{S_k^*})|} \leq \frac{D_c(G)}{n_{i-1}} \quad \text{for } i \geq 2, \tag{6}$$

and at most $1/(n_0 - n_1)$ for $i = 1$. There is something to be explained about (6).

(a) As we shall see later, only those i with $n_{i-1} - n_i > 0$ count. Hence for simplicity of statement, we assume that $n_{i-1} - n_i > 0$ for all i.

(b) As a consequence of the above assumption, S_k^* is not chosen after the $(i-1)$th iteration since choosing S_k^* covers all the elements in N_k. Furthermore, $n_0 - n_1 > 0$ implies that

$$S_k^* \text{ is cover-adjacent with } S_1. \tag{7}$$

Hence S_k^* is a candidate to be chosen as S in the ith iteration for $i \geq 2$. By the greedy choice of S_i, the first inequality of (6) holds.

(c) Also by observation (7), we have $|P_{S_k^*}| \leq D_c(G)$. Moreover, since choosing S_k^* could cover all the remaining elements in N_k, we have $|C(P_{S_k^*})| \geq n_{i-1}$. The second inequality in (6) holds.

Then by a standard analysis in dealing with set cover problem (see for example [5] Section 35.3), we have

$$\sum_{v \in N_k} w(v) \leq (n_0 - n_1)\frac{1}{n_0 - n_1} + D_c(G) \sum_{i=2}^{opt} \frac{n_{i-1} - n_i}{n_{i-1}} \leq 1 + D_c(G)(H(n_1) - H(n_{opt})).$$

Inequality (5) follows from the observation that $n_{opt} = 0$ and $n_1 < n_0 = |N_k| \leq |S_k^*| \leq \gamma$. Combining inequalities (3)–(5), we have

$$|\mathcal{R}_K| = \sum_{k=1}^{opt} \sum_{v \in N_k} w(v) \leq (1 + D_c(G)H(\gamma - 1)) \cdot opt.$$

The theorem is proved. □

4. A greedy algorithm for minimum $(2, m)$-CSC

To compute a $(2, m)$-CSC, we make use of the *ear decomposition* of 2-connected graphs. An *ear* of a graph G is a path P in G such that all internal vertices on P has degree two in G. An ear is *open* if its two ends are different, otherwise it is *closed*. A cycle is a closed ear. The ear decomposition theorem says that every 2-connected graph which is not a cycle has an open ear P such that the graph obtained by deleting internal vertices of P from G is still 2-connected. In other words, a graph G is 2-connected if and only if G can be constructed in the following way: Starting from a cycle (that is, a closed ear); iteratively adding open ears to the graph.

Algorithm 3 computes a $(2, m)$-CSC by greedy strategy. It starts from a 'most efficient' cycle, then iteratively adds 'most efficient' open ears to it until all the cover requirements are satisfied.

To compute the open ears, we use the concept of shortest (u, v)-cycle. For two distinct vertices u, v in a graph G, a *shortest* (u, v)*-cycle* is a cycle in G through u and v such that the length of the cycle (that is, the number of edges in the cycle) is minimum. A shortest (u, v)-cycle can be computed by any algorithm finding *shortest pair of disjoint paths*, since the union of a pair of disjoint (u, v)-paths is an (u, v)-cycle. The shortest pair of disjoint paths problem can be solved in polynomial time (see for example [22]).

For a subgraph H of G, a shortest open ear to H through a given vertex $v \in V(G) \setminus V(H)$ can be computed as follows: Add a new vertex s to G and connect s to every vertex in H; compute a shortest (v, s)-cycle in the extended graph; then the path obtained by deleting s from this cycle is as required.

In Algorithm 3, each element $v \in V$ is assigned a label $m(v)$ which records the remaining number of times element v is to be covered. Initially $m(v) = m$ for all v. When $m(v)$ decreases to zero, we say that the cover requirement on v is satisfied. The total number of remaining cover requirements is recorded by M. Initially $M = m|V|$. Set U is used to record the elements of V whose cover requirements have not been satisfied (note that the meaning of U here is different from that in Algorithm 2).

For an ear Q_S computed in the algorithm, we use $c(Q_S)$ to denote the number of cover requirements satisfied by adding Q_S to the currently constructed 2-connected subgraph. More concretely, for each element $v \in U$, let $m'(v)$ be the number of sets in $V(Q_S) \setminus \mathcal{R}$ which cover v, and set $\widetilde{m}(v) = \min\{m'(v), m(v)\}$. Then $\widetilde{m}(v)$ is the number of requirements newly satisfied at element v by adding Q_S, and $c(Q_S) = \sum_{v \in U} \widetilde{m}(v)$ is the total number of requirements newly satisfied by adding Q_S. Define the *efficiency* of Q_S to be

$$e(Q_S) = \frac{|V(Q_S) \setminus \mathcal{R}|}{c(Q_S)}.$$

Line 6 to 11 constructs the initial cycle and line 13 to 21 iteratively adds open ears. By the ear decomposition theorem, the output of Algorithm 3 is indeed a $(2, m)$-CSC.

Algorithm 3

Input: $(V, \mathcal{S}, G)$, where G is 2-connected and every element in V is covered by at least m sets in $\mathcal{S}$.
Output: A $(2, m)$-connected set cover $\mathcal{R}$.

1: Set $M = m|V|$, $U = V$, and $m(v) = m$ for each $v \in V$.
2: Choose $S_0 \in \mathcal{S}$ such that $|S_0|$ is maximum. $\mathcal{R} = \{S_0\}$. For each element $v \in S_0$, set $m(v) = m(v) - 1$. $M = M - |S_0|$. Remove all vertices v in U with $m(v) = 0$.
3: **if** $M = 0$ **then**
4:     Output $\mathcal{R}$.
5: **else**
6:     For each $S \in \mathcal{S} \setminus \mathcal{R}$, compute a shortest (S_1, S)-cycle Q_S.
7:     Choose S such that $e(Q_S)$ is minimum.
8:     **for** each set $R \in V(Q_S) \setminus \mathcal{R}$ **do**
9:         $\mathcal{R} = \mathcal{R} \cup \{R\}$.
10:         For each element $v \in R \cap U$, $m(v) = m(v) - 1$, $M = M - 1$, and remove v from U if $m(v) = 0$.
11:     **end for**
12: **end if**
13: **while** $M > 0$ **do**
14:     Construct a graph $\widetilde{G}$ by adding a new vertex R_0 and connect R_0 to every vertex in $\mathcal{R}$.
15:     For each $S \in \mathcal{S} \setminus \mathcal{R}$, compute a shortest (R_0, S)-cycle in $\widetilde{G}$. Let Q_S be the open ear to $G[\mathcal{R}]$ obtained by deleting R_0 from this cycle.
16:     Choose S such that $e(Q_S)$ is minimum.
17:     **for** each set $R \in V(Q_S) \setminus \mathcal{R}$ **do**
18:         $\mathcal{R} = \mathcal{R} \cup \{R\}$.
19:         For each element $v \in R \cap U$, $m(v) = m(v) - 1$, $M = M - 1$, and remove v from U if $m(v) = 0$.
20:     **end for**
21: **end while**
22: Output $\mathcal{R}$.

Next, we analyze the performance ratio of Algorithm 3 using the pair diameter $PD(G)$.

Theorem 3. *The performance ratio of Algorithm 3 is* $(PD(G) - 1)(1 + H(\gamma - 1))$, *where* $\gamma = \max\{|S| \mid S \in \mathcal{S}\}$.

Proof. The proof idea is similar to that of Theorem 2. The differences lie in dealing with the multiple covering of each element and estimating the length of added ear.

Suppose $V = \{v_1, \ldots, v_n\}$ where $n = |V|$. Duplicate each element v_i by m times. Denote by $V_i = \{v_i^{(1)}, \ldots, v_i^{(m)}\}$, where $v_i^{(1)}, \ldots, v_i^{(m)}$ are the duplicates of element v_i. Set $\mathcal{V} = \bigcup_{i=1}^{n} V_i$.

Use the notation S_i, $\mathcal{R}_k$ as in the proof of Theorem 2. Here, S_0 is the set chosen in line 2, S_1 is the set chosen in line 7. Suppose Algorithm 3 runs K rounds. For $i \geq 1$, when S_i is chosen, sets in $V(Q_{S_i}) \setminus \mathcal{R}$ are added into $\mathcal{R}$ sequentially in line 8 to line 11 for $i = 1$ an in line 17 to line 19 for $i \geq 2$. When it is the turn to deal with $R \in V(Q_{S_i}) \setminus \mathcal{R}$, a vertex $v \in R \cap U$ has its copy $v^{(m(v))}$ assigned a weight $e(Q_{S_i})$ (recall that $1 \leq m(v) \leq m$ is the remaining cover requirements on v just before R is added to $\mathcal{R}$). We may regard R as covering $v^{(m(v))}$. When $i = 0$, each element $v \in S_0$ has its copy $v^{(m)}$ assigned a weight $1/|S_0|$. Then each element $v^{(j)} \in \mathcal{V}$ is assigned a weight exactly once.

Suppose $\mathcal{R}^* = \{S_1^*, \ldots, S_{opt}^*\}$ is an optimal solution to the $(2, m)$-CSC problem. Define a partition $N_1, \ldots, N_{opt}$ of $\mathcal{V}$ as follows (write $\mathcal{N}_i = \bigcup_{k=1}^{i} N_k$ for simplicity): Set $N_1 = \{v^{(1)} \mid v \in S_1^*\}$, and for $i = 2, \ldots, opt$, set $N_i = \{v^{(j)} \mid v \in S_i^*,\ v^{(m)} \notin \mathcal{N}_{i-1},\ j$ is the first index such that $v^{(1)}, \ldots, v^{(j-1)} \in \mathcal{N}_{i-1}$ and $v^{(j)} \notin \mathcal{N}_{i-1}\}$. Fig. 1 illustrates the partition.

The remaining proof is similar to that in Theorem 2. The only difference is using $PD(G) - 1$ to upper bound $|V(Q_{S_k^*}) \setminus \mathcal{R}_i|$: since the path $Q_{S_k^*}$ has exactly two ends in $\mathcal{R}_i$, we have $|V(Q_{S_k^*}) \setminus \mathcal{R}_i| = |V(Q_{S_k^*})| - 2 \leq PD(G) - 1$. It should be noted that for each $k \in \{1, \ldots, opt\}$, $|N_k| \leq \gamma$ since each element v has at most one copy in N_k. $\square$

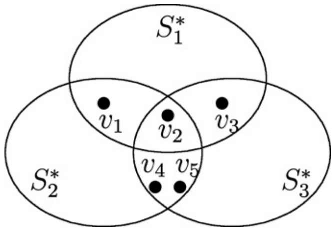

Fig. 1. An illustration of the partition. $m = 2$, $N_1 = \{v_1^{(1)}, v_2^{(1)}, v_3^{(1)}\}$, $N_2 = \{v_1^{(2)}, v_2^{(2)}, v_4^{(1)}, v_5^{(1)}\}$, $N_3 = \{v_3^{(2)}, v_4^{(2)}, v_5^{(2)}\}$.

5. Discussion

In this paper, we gave two approximation algorithms for Minimum Connected Set Cover problem in general graphs. Logarithm performance guarantee was obtained, incorporating a new parameter D_c which measures the maximum distance between two sets covering a common element. We also gave a logarithm approximation algorithm for Minimum $(2, m)$-Connected Set Cover problem, using a new parameter $PD(G)$ which in fact measures the maximum length of an ear. These are the first algorithms for CSC problems in general graphs with guaranteed performance ratio. To improve the performance ratio is one of our future directions. To study the Minimum (k, m)-CDS problem for $k \geq 3$ is another direction. a weighted version of the CSC problem is also an interesting topic. However, the methods used in this paper cannot be generalized for that. A lot of deep insight and new ideas are needed to solve it.

Acknowledgements

The research of the first author is supported by NSFC (60603003), the Key Project of Chinese Ministry of Education (208161) and Scientific Research Program of the Higher Education Institution of XinJiang. The work of the third author is supported in part by the National Science Foundation under grant CCF-0514796 and CNS-0524429.

References

[1] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi, Complexity and Approximation, Springer, 2003.
[2] R.A. Briers, Incorporating connectivity into reserve selection procedures, Biological Conservation 103 (2002) 77–83.
[3] J.O. Cerdeira, L.S. Pinto, Requiring connectivity in the set covering problem, Journal of Combinatorial Optimization 9 (2005) 35–47.
[4] D.H. Chen, D.Z. Du, X.D. Hu, G.H. Lin, L.S. Wang, G.L. Xue, Approximations for Steiner trees with minimum number of Steiner points, Theoretical Computer Science 262 (2001) 83–99.
[5] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, second ed..
[6] D.M. Debinski, R.D. Holt, A survey and overview of habitat fragmentation experiments, Conservation Biology 14 (2000) 342–355.
[7] D.Z. Du, L.S. Wang, B.G. Xu, The Euclidean bottleneck Steiner tree and Steiner tree with minimum number of Steiner points, in: Computing and Combinatorics (Guilin, 2001), in: Lecture Notes in Computer Science, vol. 2108, Springer, Berlin, 2001, pp. 509–518.
[8] U. Feige, A threshold of ln *n* for approximating set cover, in: Proc. 28th ACM Symposium on Theory of Computing, ACM, 1996, pp. 314–318.
[9] M.R. Garey, D.S. Johnson, Computers and Intractability, W.H. Freeman and Company, New York, 1979.
[10] S. Guha, S. Khuller, Improved methods for approximating node weighted Steiner trees and connected dominating sets, Information and Computation 150 (1999) 57–74.
[11] P. Klein, R. Ravi, A nearly best-possible approximation algorithm for node-weighted Steiner trees, Journal of Algorithms 19 (1995) 104–115.
[12] G.H. Lin, G. Xue, Steiner tree problem with minimum number of Steiner points and bounded edge-length, Information Processing Letters 69 (1999) 53–57.
[13] I.I. Măndoiu, A.Z. Zelikovsky, A note on the MST heuristic for bounded edge-length Steiner trees with minimum number of Steiner points, Information Processing Letters 75 (2000) 165–167.
[14] M.D. McDonnell, H.P. Possingham, I.R. Ball, E.A. Cousins, Mathematical methods for spatially cohesive reserve design, Environmental Modeling and Assessment 7 (2002) 107–114.
[15] D.J. Nalle, J.L. Arthur, J. Sessions, Designing compact and contiguous reserve networks ith a hybrid heuristic approach, Forest Science 48 (2002) 59–68.
[16] A.O. Nicholls, C.R. Margules, An upgraded reserve selection algorithm, Biological Conservation 64 (1993) 165–169.
[17] H. Önal, R.A. Briers, Incorporating spatial criteria in optimum reserve network selection, Proceedings of the Royal Society, London B 269 (2002) 2437–2441.
[18] H. Önal, R.A. Briers, Selection of a minimum-boundary reserve network using integer programming, Proceedings of the Royal Society, London B 270 (2003) 1487–1491.
[19] H. Possingham, I. Ball, S. Andelman, Mathematical methods for identifying representative reserve networks, in: S. Ferson, M. Burgman (Eds.), Quantitative Methods for Conservation Biology, Springer, New York, 2000, pp. 291–306.
[20] T.P. Shuai, X.D. Hu, Connected set cover problem and its applications, in: International conference on algorithmic aspects in information and management, Hong Kong, China, in: Lecture Notes in Computer Science, vol. 4041, 2006, pp. 243–254.
[21] P. Siitonen, A. Tanskanen, A. Lehtinen, Selecting forest reserves with a multi-objective spatial algorithm, Environmental Science and Policy 6 (2003) 301–309.
[22] J.W. Suurballe, R.E. Tarjan, A quick method for finding shortest pairs of disjoint paths, Networks 14 (1984) 325–336.
[23] F. Zou, X.Y. Li, D. Kim, W.L. Wu, Two constant approximation algorithms for node-weighted Steiner tree in nnit disk graphs, in: 2nd Annual International Conference on Combinatorial Optimization and Application, St. John's, Newfoundland, 2008.