Theoretical Computer Science 535 (2014) 54–58

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Note

Approximation algorithm for the minimum weight connected k-subgraph cover problem ☆

Yaping Zhang [a], Yishuo Shi [a], Zhao Zhang [b,*]

[a] College of Mathematics and System Sciences, Xinjiang University, Urumqi, Xinjiang, 830046, People's Republic of China
[b] College of Mathematics Physics and Information Engineering, Zhejiang Normal University, Zhejiang, Jinhua, 321004, China

ARTICLE INFO

Article history:
Received 16 December 2013
Received in revised form 5 March 2014
Accepted 24 March 2014
Communicated by P. Widmayer

Keywords:
k-Path vertex cover
Connected k-subgraph vertex cover

ABSTRACT

A subset F of vertices is called a connected k-subgraph cover (VCC_k) if every connected subgraph on k vertices contains at least one vertex from F. The minimum weight connected k-subgraph cover problem ($MWVCC_k$) has its background in the field of security and supervisory control. It is a generalization of the minimum weight vertex cover problem, and is related with the minimum weight k-path cover problem ($MWVCP_k$) which requires that every path on k vertices has at least one vertex from F. A k-approximation algorithm can be easily obtained by LP rounding method. Assuming that the girth of the graph is at least k, we reduce the approximation ratio to $k-1$, which is tight for our algorithm.

1. Introduction

The topology of a wireless sensor network (WSN) can be modeled as a graph, in which vertices represent sensors and edges represent communication channels between sensors. In recent years, new security protocols for WSN emerge. For example, in the k-generalized Canvas scheme [12] which guarantees data integrity, two kinds of sensor devices, protected and unprotected, are distinguished. An attacker is unable to copy data from a protected device. Suppose each information can be stored in a path of k vertices. So, it is required that every such a path has at least one protected vertex. The problem is to minimize the cost of the network by minimizing the number of protected vertices. Such a consideration leads to the *minimum weight k-path vertex cover problem* ($MWVCP_k$), the goal of which is to find a minimum weight vertex set F such that every path of k vertices contains at least one vertex from F.

In this paper, we propose a related problem as follows: Given a graph $G=(V,E)$ and a vertex-weight function w, the goal is to find a minimum weight vertex set $F\subseteq V$ such that every connected subgraph on k vertices has at least one vertex from F. Call such a set F as a *connected k-subgraph cover* (VCC_k) and the problem as a *minimum weight connected k-subgraph cover* problem ($MWVCC_k$). For $k=2$, $MWVCC_2$ is exactly the minimum weight vertex cover problem. For $k=3$, $MWVCC_3$ is the same as $MWVCP_3$.

This problem also has its background in the field of security and supervisory control. For example, in a WSN, if an attacker knows at least k related information fragments, then he can decode the whole information. Therefore, every connected k-vertex set must have at least one protected vertex to ensure security. For another example, if every k connected

☆ This research is supported by NSFC (61222201), SRFDP (20126501110001), Xingjiang Talent Youth Project (2013711011), and Xinjiang Graduate Research Project (XJGRI2013012).
* Corresponding author.
E-mail address: hxhzz@sina.com (Z. Zhang).

http://dx.doi.org/10.1016/j.tcs.2014.03.026
0304-3975/© 2014 Elsevier B.V. All rights reserved.

sensors can work as a group, then in order to control their work, at least one sensor from every potential work group should be supervised.

It is not difficult to obtain a k-approximation for $MWVCC_k$ (as well as for $MWVCP_k$), using LP rounding technique. Under the assumption that the girth (the length of a shortest cycle) is at least k, we improve the approximation ratio for $MWVCC_k$ to $k-1$. Factor $k-1$ is tight for our algorithm.

In [14], Tu and Zhou gave a 2-approximation algorithm for $MWVCP_3$. Since $MWVCP_3$ is the same as $MWVCC_3$ and the girth of a simple graph is always at least three, Tu and Zhou's result [14] is included in our result.

The remainder of this paper is organized as follows. We first introduce some related works in Section 2. In Section 3, we present our algorithm and its theoretical analysis. In Section 4, we conclude the paper with a discussion on future work.

2. Related work and preliminaries

Related work in this section is focused on approximation results on Minimum k-Path Vertex Cover problem ($MVCP_k$) and Minimum Weight k-Path Cover problem ($MWVCP_k$).

The $MVCP_k$ problem was proposed in [12]. In [5], Bresar et al. gave a polynomial-time approximation-preserving reduction from the Minimum Vertex Cover problem to $MVCP_k$, which, combining with [6], implies that for every $k \ge 2$, $MVCP_k$ is not able to be approximated within a factor of 1.3606 unless P = NP. They also gave a linear-time algorithm for $MVCP_k$ on trees and some upper bounds on the minimum cardinality of VCP_k. In [9], Kardoš et al. presented a polynomial-time randomized approximation algorithm for VCP_3 with an expected approximation ratio 23/11. They also formulated as an open problem whether $MVCP_k$ has a constant approximation for each $k \ge 2$. It was proved by Tu et al. [13] that $MVCP_3$ is NP-hard even for a cubic planar graph of girth 3, and a 1.57-approximation greedy algorithm was given for VCP_3 in cubic graphs. Recently, Li and Tu [10] presented a 2-approximation for VCP_4 in cubic graphs.

Requiring that the k-path vertex cover induces a connected subgraph, the problem is the Minimum k-Path Connected Vertex Cover ($MCVCP_k$). In [11], Liu et al. gave a PTAS for $MVCP_k$ in unit disk graphs.

The above works mainly concentrate on unweighted VCP_k problem.

Considering weight, Tu and Zhou [14] gave a 2-approximation for $MWVCP_3$ by using a layering method. By using a primal–dual method, they also achieved a 2-approximation [15].

For general k, it is not difficult to obtain a k-approximation for $MWVCP_k$ as well as for $MWVCC_k$. In fact, $MWVCC_k$ can be modeled as the following integer linear program:

$$\begin{aligned} \min\ & \sum_{i=1}^{n} w_i x_i \\ \text{s.t.}\ & \sum_{i\in S} x_i \ge 1, \quad \forall S \subseteq V,\ |S| = k,\ G[S] \text{ is connected}, \\ & x_i \in \{0,1\}, \quad i = 1, 2, \cdots, n, \end{aligned} \tag{1}$$

where $G[S]$ is the subgraph of G induced by vertex set S. By a classical rounding technique (see, for example [16]), one has a k-approximation. To be more concrete, solving the relaxed linear program of (1) (that is, relax $x_i \in \{0,1\}$ to $0 \le x_i \le 1$) to obtain an optimal fractional solution x^*. Let $x_i^A = 1$ if $x_i^* \ge \frac{1}{k}$, and $x_i^A = 0$ otherwise. Then $C = \{v_i : x_i^A = 1\}$ is a VCC_k of G and $w(C) = \sum_{i=1}^n w_i x_i^A \le k \sum_{i=1}^n w_i x_i^* = k \cdot opt^f \le k \cdot opt$, where opt^f is the optimal fractional value for the relaxation of (1) and opt is the optimal integral value for (1). The $MWVCP_k$ problem can be modeled by a similar 0–1 integer linear program as (1), except that "$G[S]$ is connected" is replaced by "$G[S]$ is a path on k vertices".

In this paper, we present a $(k-1)$-approximation for $MWVCC_k$ under the assumption that the girth of G is at least k, using local ratio method. Local ratio method was first proposed by Bar-Yehuda and Even [3], and has been used to design approximation algorithms for the feedback vertex set problem [1], the node deletion problem [8], resource allocation and scheduling problems [2], the minimum s–t cut problem and the assignment problem [4]. The readers may refer to [7] for a systematic introduction of the local ratio method. The key step in obtaining the desired approximation ratio is to find a special weight function w_1 and prove the desired approximation ratio with respect to w_1. In the following section, we shall put our focus on how to realize this step for $MWVCC_k$, and how to make use of such w_1 recursively.

3. The algorithm and its theoretical analysis

Let $d_G(v)$ denote the degree of vertex v in G. The subscript G is omitted if there is no ambiguity in the context. Given a vertex subset S, let $E[S]$ denote the set of edges having both ends in S, and let $G[S]$ denote the subgraph of G induced by S. Notice that $F \subseteq V$ is a VCC_k if every component of $G[V \setminus F]$ has cardinality at most $k-1$. A VCC_k F is said to be *minimal* if for any $v \in F$, $F - \{v\}$ is no longer a VCC_k. Let γ denote the size of a VCC_k of G with the smallest cardinality.

Theorem 3.1. *Let G be a connected graph, k be an integer with $k \ge 3$, and F be a VCC_k. Suppose the girth of G, denoted as $g(G)$, is at least k. Then*

$$\sum_{v\in F} (k-1)d(v) \ge (k-1)|E| - (k-2)|V| + (k-2)\gamma. \tag{2}$$

Furthermore, if F is a minimal VCC$_k$, then

$$\sum_{v\in F} d(v) \le (k-1)|E| - (k-2)|V| + (k-2)\gamma. \tag{3}$$

Proof. First, we prove inequality (2). If $F = V$, then inequality (2) holds trivially, since $\sum_{v\in V} d(v) = 2|E|$ and $\gamma \le |V|$. Now, suppose $F \ne V$. Since $g(G) \ge k$ and every component of $G[V \setminus F]$ has cardinality at most $k-1$, every component of $G[V \setminus F]$ is a tree. So

$$\big|E[V \setminus F]\big| = |V \setminus F| - t, \tag{4}$$

where t is the number of connected components in $G[V \setminus F]$. It follows that

$$|E| \le \sum_{v\in F} d(v) + \big|E[V \setminus F]\big| = \sum_{v\in F} d(v) + |V| - |F| - t.$$

By observing that $t \ge \frac{|V|-|F|}{k-1}$, we obtain

$$\sum_{v\in F} d(v) \ge |E| - \frac{k-2}{k-1}\big(|V| - |F|\big).$$

Then inequality (2) follows from $|F| \ge \gamma$.

We now prove inequality (3). Notice that the righthand side of inequality (3) can be rewritten as

$$\begin{aligned}&\sum_{v\in V} d(v) + (k-3)|E| - (k-2)|V| + (k-2)\gamma \\ &\quad= \left(\sum_{v\in F} d(v) + 2\big|E[V \setminus F]\big| + \big|\delta(F)\big|\right) + (k-3)|E| - (k-2)|V| + (k-2)\gamma,\end{aligned} \tag{5}$$

where $\delta(F)$ is the set of edges with one end in F and the other end in $V \setminus F$. Combining $|E| = |E[F]| + |E[V \setminus F]| + |\delta(F)|$ with (4) and (5), we see that proving inequality (3) is equivalent to proving the following:

$$(k-2)\big|\delta(F)\big| \ge (k-2)|F| + (k-2)t - \big|E[V \setminus F]\big| - (k-3)\big|E(F)\big| - (k-2)\gamma. \tag{6}$$

Observing that for each $v \in F$, there is a connected subgraph C_v on k vertices such that $C_v \cap F = \{v\}$, otherwise F would not be minimal. Call such a C_v as a *witness* of v. Let $\mathcal{C} = \{C_v : v \in F\}$ and T be a maximum sub-collection of $\mathcal{C}$ such that witnesses in T are mutually vertex-disjoint. Let $F_T = \{v \in F : C_v \in T\}$ and $\tilde{F} = F \setminus F_T$. Notice that $|F| = |T| + |\tilde{F}|$, $|T| \le \gamma$, and $k \ge 3$. So, to prove (5), it suffices to prove

$$\big|\delta(F)\big| \ge |\tilde{F}| + t. \tag{7}$$

For every component H in $G[V \setminus F]$, there is an edge between H and F because G is connected. Choose such an edge to correspond to H, and denote it as e_H. Furthermore,

$$\text{if it is possible, then choose } e_H \text{ to be an edge between } H \text{ and } F_T. \tag{8}$$

Next, we shall prove that each vertex $v \in \tilde{F}$ is incident with an edge $e_v \in \delta(F)$ such that $e_v \ne e_H$ for any component H of $G[V \setminus F]$. For this purpose, notice that by the maximality of T, there is a vertex $w \in F_T$ such that $V(C_w) \cap V(C_v) \ne \emptyset$. Since v is the only vertex of C_v in F and w is the only vertex of C_w in F, we see that there is a component H in $G[V \setminus F]$ which is adjacent with both v and w. By the choice of e_H (see (8)), the edge between v and H can serve as the desired e_v. It follows that edges in $\{e_H\}_{H \text{ is a component of } G[V\setminus F]} \cup \{e_v\}_{v\in\tilde{F}}$ are all distinct, and thus $|\delta(F)| \ge |\{e_H\}| + |\{e_v\}| = |\tilde{F}| + t$. This finishes the proof. □

In the following, we give a $(k-1)$-approximation algorithm for *MWVCC$_k$*. For this purpose, we first consider a special vertex weight function w_1 called a *degree-weight function*, that is, $w_1(v) = c \cdot d(v)$ ($\forall v \in V$) for some constant c.

Lemma 3.2. *Let w_1 be a degree-weight function on the vertices of $G = (V, E)$, F be a minimal VCC$_k$ of G and F^* be a minimum weight VCC$_k$ of G. Then $w_1(F) \le (k-1) \cdot w_1(F^*)$.*

Proof. We may assume that G is connected. By Theorem 3.1, we have

$$w_1(F) = \sum_{v\in F} w_1(v) = c \cdot \sum_{v\in F} d(v) \le c \cdot (k-1) \sum_{v\in F^*} d(v) = (k-1) w_1\big(F^*\big).$$

The lemma is proved. □

For a general nonnegative weight function w, we can recursively decompose it into degree-weight functions, which are denoted as $t_0, t_1, \cdots, t_l$ in Algorithm 1. Algorithm 1 constructs a nested sequence of subgraphs $G = H_0 \supset H_1 \supset H_2 \cdots \supset H_l$, where H_i is obtained from H_{i-1} by removing vertices of residual weight zero, i.e., every vertex $v \in V(H_{i-1}) \setminus V(H_i)$ has $w^i(v) = 0$. Since components in H_i with cardinality less than k play no role in a minimal VCC_k, so the algorithm continues to work on G_i, which is obtained from H_i by removing such components. Since $V(G_l) = \emptyset$, every component in H_l has cardinality smaller than k, and thus $F_l = \emptyset$ is a minimum VCC_k of H_l. Then the algorithm extends it recursively in a backward manner to a VCC_k of $G_0 = G$. It should be noticed that $F_i \cup (V(G_{i-1}) \setminus V(H_i))$ is a VCC_k of G_{i-1}, because $G_{i-1} - (F_i \cup (V(G_{i-1}) \setminus V(H_i))) = H_i - F_i = (G_i - F_i) \cup (H_i - G_i)$, and every component of $G_i - F_i$ and $H_i - G_i$ has at most $k-1$ vertices. So, a vertex set V_i as in Line 11 of Algorithm 1 exists.

Algorithm 1 Algorithm for *MWVCC*$_k$.

Input: A connected vertex-weighted graph G and a positive integer k.
Output: A VCC_k F_0.
1: $l \leftarrow 0,\ G_0 = G,\ w^l \leftarrow w$.
2: **while** $|V(G_l)| \neq \emptyset$ **do**
3: $\quad c \leftarrow \min_{u \in V(G_l)} \left\{ \frac{w^l(u)}{d_{V(G_l)}(u)} \right\}$.
4: $\quad$For each vertex $v \in V(G_l)$, $t_l(v) \leftarrow c \cdot d_{G_l}(v)$, $w^{l+1}(v) \leftarrow w^l(v) - t_l(v)$.
5: $\quad l \leftarrow l + 1$.
6: $\quad H_l \leftarrow$ the subgraph of G_{l-1} induced by vertices v with $w^l(v) > 0$.
7: $\quad G_l \leftarrow$ the union of those components of H_l with at least k vertices.
8: **end while**
9: $F_l \leftarrow \emptyset$.
10: **for** $i = l, l-1, \cdots, 1$ **do**
11: $\quad$Choose a minimal set of vertices V_i from $V(G_{i-1}) \setminus V(H_i)$ such that $F_{i-1} = F_i \cup V_i$ is a VCC_k of G_{i-1}.
12: **end for**

Theorem 3.3. *The running time of Algorithm 1 is* $O(|V|^2|E|)$.

Proof. The running time follows from the observation that both the while loop and the for loop are executed at most $|V|$ times, and each while loop needs time $O(|V| + |E|)$, each for loop needs time $O(|V| \cdot |E|)$. For example, Line 11 can be accomplished by setting $V_i = \emptyset$ initially, and checking vertices $v \in V(G_{i-1}) \setminus V(H_i)$ one by one. As long as adding v to $G_{i-1} \setminus (F_i \cup V_i)$ creates a component of cardinality at least k, then set $V_i := V_i \cup \{v\}$. Since checking whether $(G_{i-1} \setminus (F_i \cup V_i)) \cup \{v\}$ has a component of cardinality at least k requires time $O(|E|)$, Line 11 can be done in time $O(|V| \cdot |E|)$. □

Theorem 3.4. *Algorithm 1 is a* $(k-1)$*-approximation for MWVCC*$_k$.

Proof. We show by induction on i that F_i is a minimal VCC_k of G_i and is a $(k-1)$-approximation for $MWVCC_k$ on G_i with respect to weight function w^i. This is trivially true for $i = l$.

Suppose the claim is true for F_i, and $V_i \subseteq V(G_{i-1}) \setminus V(H_i)$ is a minimal vertex set such that $F_{i-1} = F_i \cup V_i$ is a VCC_k of G_{i-1}. We first show that $F_{i-1} = F_i \cup V_i$ is a *minimal* VCC_k of G_{i-1} (this implies that a minimal VCC_k of G_{i-1} can be obtained by extending F_i while keeping vertices in F_i intact). To see this, it suffices to show that for any vertex $v \in F_{i-1}$, $G_{i-1} - (F_{i-1} \setminus \{v\})$ has a connected subgraph on k vertices. This is true for $v \in V_i$ because of the minimality of V_i. For $v \in F_i$, by the minimality of F_i, $G_i - (F_i \setminus \{v\})$ has a connected subgraph of k vertices, and so has $G_{i-1} - (F_{i-1} \setminus \{v\})$, because $G_i - (F_i \setminus \{v\})$ is a subgraph of $G_{i-1} - (F_{i-1} \setminus \{v\})$.

Let F^*_{i-1} be a minimum VCC_k of G_{i-1} with respect to weight function w^{i-1}. Notice that $F^*_{i-1} \cap V(G_i)$ is a VCC_k of G_i. So by induction hypothesis,

$$w^i(F_i) \le (k-1) w^i\big(F^*_{i-1} \cap V(G_i)\big). \tag{9}$$

By Line 6 of Algorithm 1, $w^i(v) = 0$ for any $v \in V(G_{i-1}) \setminus V(H_i)$. So by Line 11 of Algorithm 1,

$$w^i(V_i) = 0. \tag{10}$$

Combing (9), (10) with $F_{i-1} = F_i \cup V_i$, we have

$$w^i(F_{i-1}) = w^i(F_i) \le (k-1) w^i\big(F^*_{i-1} \cap V(G_i)\big) \le (k-1) w^i\big(F^*_{i-1}\big). \tag{11}$$

Since F_{i-1} is a *minimal* VCC_k of G_{i-1} and t_{i-1} is a degree weight function on $V(G_{i-1})$, by Lemma 3.2,

$$t_{i-1}(F_{i-1}) \le (k-1) t_{i-1}\big(F^*_{i-1}\big). \tag{12}$$

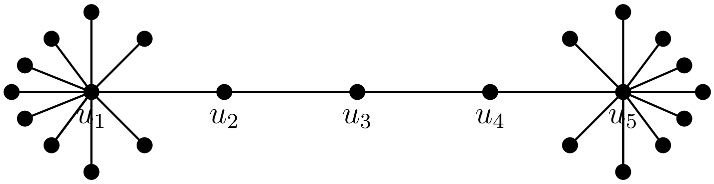

Fig. 1. For a degree-weight function, minimal VCP_k does not approximate the optimal solution within a constant factor for $k \geq 4$.

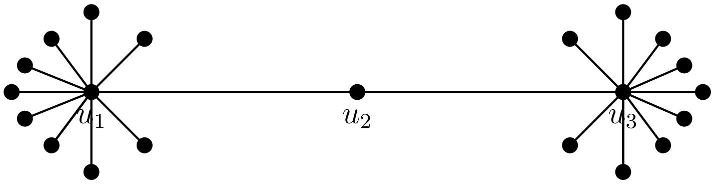

Fig. 2. Tight example for our algorithm.

Summing (11), (12) together, and noticing that $w^{i-1}(v) = w^i(v) + t_{i-1}(v)$ for $v \in V(G_{i-1})$ (by Line 4 of Algorithm 1), we have $w^{i-1}(F_{i-1}) \leq (k-1)w^{i-1}(F^*_{i-1})$. This finishes the proof for the induction step.

In particular, taking $i = 0$, F_0 is a VCC_k of $G_0 = G$ which approximates F^*_0 within a factor of $k-1$. □

4. Discussion and future work

Tu and Zhou proved in [14] that for a degree-weight function, any minimal VCP_3 approximates an optimal VCP_3 within a factor of 2. This cannot be generalized to $MWVCP_k$ with $k \geq 4$. Consider the example in Fig. 1, where $k = 4$. Suppose the weight function $w(u) = d(u)$ for $u \in V(G)$, and $d(u_1) = d(u_5) = \Delta$, where Δ is the maximum degree of the graph. Then $\{u_3\}$ is the minimum VCP_4 with weight 2, and $\{u_1, u_5\}$ is a minimal VCP_4 with weight 2Δ. This implies that for $MWVCP_k$ with $k \geq 4$, taking minimal VCP_k cannot achieve a constant approximation.

In this paper, we obtain a $(k-1)$-approximation for $MWVCC_k$ when the girth is at least k. The factor $k-1$ is tight in the following sense. Consider the example in Fig. 2. Suppose the weight function $w(u) = d(u)$ for $u \in V(G)$, and $d(u_1) = d(u_3) = k-1$. Then $\{u_2\}$ is the optimal solution of VCC_k with weight 2, and $\{u_1, u_3\}$ is a minimal VCC_k with weight $2(k-1)$, which is $(k-1)$ times the optimal value.

As a future work, we are interested in achieving better approximation for $MWVCC_k$ in some special graphs emerging from WSN, such as disk graphs. And for general graphs, better approximation without girth assumption needs to be further explored.

References

[1] V. Bafna, P. Berman, T. Fujito, A 2-approximation algorithm for the undirected feedback vertex set problem, SIAM J. Discrete Math. 12 (1999) 289–297.
[2] A. Bar-Noy, R. Bar-Yehuda, A. Freund, J. Naor, B. Shieber, A unified approach to approximating resource allocation and scheduling, J. Assoc. Comput. Math. 48 (2001) 1069–1090.
[3] R. Bar-Yehuda, S. Even, A local-ratio theorem for approximating the weighted vertex cover problem, Ann. Discrete Math. 25 (1985) 27–46.
[4] R. Bar-Yehuda, D. Rawitz, Local ratio with negative weights, Oper. Res. Lett. 32 (2004) 540–546.
[5] B. Bresar, F. Kardos, J. Katrenic, G. Semanisin, Minimum k-path vertex cover, Discrete Appl. Math. 159 (2011) 1189–1195.
[6] I. Dinur, S. Safra, On the hardness of approximating minimum vertex cover, Ann. of Math. (2) 162 (2005) 439–485.
[7] D.-Z. Du, K.-I. Ko, X. Hu, Design and Analysis of Approximation Algorithms, Springer, New York, 2012.
[8] T. Fujito, A unified approximation algorithm for node-deletion problems, Discrete Appl. Math. 86 (1998) 213–231.
[9] F. Kardoš, J. Katrenič, I. Schiermeyer, On computing the minimum 3-path vertex cover and dissociation number of graphs, Theoret. Comput. Sci. 412 (2011) 7009–7017.
[10] Y. Li, J. Tu, A 2-approximation algorithm P_4 for the vertex cover problem in cubic graphs, Int. J. Comp. Math. (2014), http://dx.doi.org/10.1080/00207160.2014.881476 (published online).
[11] X. Liu, H. Lu, W. Wang, W. Wu, PTAS for the minimum k-path connected vertex cover problem in unit disk graphs, J. Global Optim. (2013) 449–458.
[12] M. Novotny, Design and analysis of a generalized canvas protocol, in: Proceedings of WISTP 2010, in: LNCS, vol. 6033, 2010, pp. 106–121.
[13] J.H. Tu, F. Yang, The vertex cover P_3 problem in cubic graphs, Inform. Process. Lett. 113 (2013) 481–485.
[14] J.H. Tu, W. Zhou, A factor 2 approximation algorithm for the vertex cover P_3 problem, Inform. Process. Lett. 111 (2011) 683–686.
[15] J.H. Tu, W. Zhou, A primal–dual approximation algorithm for the vertex cover P_3 problem, Theoret. Comput. Sci. 412 (2011) 7044–7048.
[16] V.V. Vazirani, Approximation Algorithms, Springer, Berlin, Heidelberg, New York, 2001.