Theoretical Computer Science 571 (2015) 58–66

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

A PTAS for the minimum weight connected vertex cover P_3 problem on unit disk graphs ☆

Limin Wang [a], Xiaoyan Zhang [a,b,*], Zhao Zhang [c], Hajo Broersma [b]

[a] School of Mathematical Science & Institute of Mathematics, Nanjing Normal University, Nanjing, 210023, China
[b] Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
[c] College of Mathematics Physics and Information Engineering, Zhejiang Normal University, Zhejiang, Jinhua, 321004, China

ARTICLE INFO

Article history:
Received 3 July 2014
Received in revised form 24 November 2014
Accepted 10 January 2015
Available online 15 January 2015
Communicated by P. Widmayer

Keywords:
PTAS
Vertex cover
P_3 cover
Minimum weight cover
Connected cover
c-Local problem
Unit disk graph
Grid graph

ABSTRACT

Let $G = (V, E)$ be a weighted graph, i.e., with a vertex weight function $w : V \to \mathbb{R}^+$. We study the problem of determining a minimum weight connected subgraph of G that has at least one vertex in common with all paths of length two in G. It is known that this problem is NP-hard for general graphs. We first show that it remains NP-hard when restricted to unit disk graphs. Our main contribution is a polynomial time approximation scheme for this problem if we assume that the problem is c-local and the unit disk graphs have minimum degree of at least two.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

We use Bondy and Murty [1] for standard graph theory and Garey and Johnson [5] for standard computational complexity terminology and notation.

The vertex cover problem (VC) is identified as one of the six basic NP-complete (decision) problems in Section 3.1 of [5] and has been a popular topic since. Given a graph $G = (V, E)$ and a positive integer $k \le |V|$, the question in VC is whether there exists a vertex cover(ing) of size at most k in G, i.e., a subset $S \subseteq V$ such that $|S| \le k$ and every edge of E has an end vertex in S. Many problems have been shown to be NP-complete by a reduction (called transformation in [5]) from VC, including well-known problems as the Hamilton cycle problem and the clique problem. We refer the interested reader to [5] for definitions and details. Vertex covers are also fundamental within graph theory, one reason being that (minimum) vertex covers can be considered as the duals of (maximum) matchings. The VC problem also represents a large class of related vertex deletion problems, in which one is interested in determining a (preferably small) subset $S \subseteq V$ such that $G - S$ has a desired property, e.g., such that $G - S$ is edgeless (S is a vertex cover) or $G - S$ is a forest (S is a feedback

☆ This research is supported by NSFC (11471003, 61222201), SRFDP (20126501110001), Xinjiang Talent Youth Project (2013711011), PAPD and the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (13KJB1100188).

* Corresponding author at: School of Mathematical Science & Institute of Mathematics, Nanjing Normal University, Nanjing, 210023, China.
E-mail addresses: wanglimin0721@163.com (L. Wang), royxyzhang@gmail.com, zhangxiaoyan@njnu.edu.cn (X. Zhang), hxhzz@sina.com (Z. Zhang), h.j.broersma@utwente.nl (H. Broersma).

http://dx.doi.org/10.1016/j.tcs.2015.01.005
0304-3975/© 2015 Elsevier B.V. All rights reserved.

vertex set). There are many other examples, including weighted variants (in which one is interested in a set S with a small total weight) and variants in which one imposes structural conditions on S (like being connected). Many of these variants are motivated by real world applications. We refer to [1] for more background and details.

1.1. The minimum weight connected vertex cover P_3 problem

Here we focus on a variant of VC in which the set $S \subseteq V$ should intersect (cover) every path of length two in G, induce a connected subgraph $G[S]$ in G, and be of minimum weight subject to these two conditions. So the graph $G - S$ in this case will consist of isolated vertices and isolated edges only. To describe more background and related recent results on this topic, we first define the more general problem.

Given a graph $G = (V, E)$ with vertex weight function $w : V \to \mathbb{R}^+$, the minimum weight vertex cover P_k problem is the problem of finding a minimum weight vertex cover set $S \subseteq V$ such that the graph $G[V \setminus S]$ has no P_k as a subgraph, where P_k is the path on k vertices. Or, equivalently, the problem is to find a minimum weight vertex cover set $S \subseteq V$ such that for any P_k in G, $V(P_k) \cap S \neq \emptyset$. Here, the weight of a vertex set $S \subseteq V$ is $w(S) = \sum_{v \in S} w(v)$.

For any vertex $v \in V$, if $v \in V(P_k)$, we say that P_k is covered by v. The set S with the property that for any P_k, $V(P_k) \cap S \neq \emptyset$ is called a vertex cover P_k set. The case $k = 2$ of this problem is the well-studied minimum weight vertex cover problem. In this paper, we consider the case that $k = 3$.

Although we study the problem from a theoretical perspective, it is not difficult to put forward some applications. One such application comes from the real world of traffic management. Increasing traffic can result in more and more traffic accidents, and this is one of the reasons why in many countries cameras are installed to monitor and control the traffic flows. Installing (one or more) cameras at every traffic junction is very costly. Therefore, one might consider the decision to install cameras at certain junctions, ensuring, e.g., that any driver will encounter at least one camera within k successive junctions, and, at the same time, at the lowest total cost. Such a decision problem in its easiest form can be modelled as a minimum weight vertex cover P_k problem.

If furthermore, the subgraph $G[S]$ of G induced by a vertex cover P_k set S is required to be connected, then we call S a connected vertex cover P_k set. The Minimum Weight Connected Vertex Cover P_3 problem (MWCVCP3 for short) is the optimisation problem of finding such a set with minimum total weight. Connectivity constraints come in naturally, e.g., in many applications concerning wireless sensor networks, in which it is usually important to ensure connectivity if the sensor devices have limited capabilities of computation, energy and communication. Moreover, they are often deployed in accessible areas, where they can be rather easily captured by attackers. Therefore, the design of security protocols has become a challenge. One such protocols, known as the Canvas protocol, was designed in [9,10] to provide data integrity or data origin authentication [9] in sensor networks. The k-generalised Canvas scheme [10] guarantees data integrity if at least one vertex is not captured on each path of length $k - 1$ in the communication graph. Thus, during the deployment and initialisation of a sensor network, it should be ensured that at least one protected vertex exists on each path of length $k - 1$ in the communication graph, and the problem of minimising the cost of the network by minimising the number of protected vertices arises naturally in [10].

1.2. Unit disk graphs

As with many, if not all of the variants of VC, the MWCVCP3 problem is NP-hard for general graphs. In fact, we will show that this problem remains NP-hard when restricted to unit disk graphs, or even to the more specific subclass of grid graphs. Fortunately, the structure of unit disk graphs allows good approximate solutions to many optimisation problems, including MWCVCP3, as we will show later.

Unit disk graphs are often used in applications involving wireless networks, as the disks can be used to represent the areas covered by the devices, and the edges of the graph can model conflicts, like interference caused by overlapping areas, or the possibility of communication between devices with overlapping areas. As the name indicates, a unit disk graph is the intersection graph of a set of unit disks in the Euclidean plane; each vertex of the graph represents the central point of a unit disk with a diameter (or radius) of unit length, say with diameter 1. An edge exists between two vertices if and only if the corresponding disks intersect (or touch), i.e., two vertices representing the central points u and v are adjacent if and only if $|uv| \leq 1$, where $|uv|$ is the Euclidean distance between u and v. The list of central points of the disks is referred to as the geometric representation of the graph. Since computing a possible geometric representation for a given unit disk graph is NP-hard [7], in this paper we assume the geometric representation of the unit disk graph is given. This is usually the case in applications.

1.3. Polynomial-time approximation scheme

Despite the fact that we are going to prove that MWCVCP3 is NP-hard for unit disk graphs, we are also going to show that it is possible to design a polynomial algorithm that finds a reasonable solution for MWCVCP3, i.e., arbitrarily close to an optimum solution. The running time of this algorithm depends on how close we want the solution to be to an optimum solution, in the following sense. A polynomial-time approximation scheme (PTAS) is a family of approximation algorithms with performance ratio $1 + \varepsilon$ (for any positive real number ε) that can be executed in polynomial time (depending on ε).

Our results are partly based on the ideas used by Zhang, Gao and Wu [14] who presented a PTAS for the minimum connected vertex cover problem for unit disk graphs.

We also need a concept that is analogous to the concept of a c-local problem, as it first appeared in a paper of Wang and Jiang [13], in which they studied Steiner tree problems in the plane with Euclidean and rectilinear metrics. They call a Steiner tree problem c-local (for some positive constant c) if in a minimum spanning tree for the terminals, the length of a longest edge is at most c times the length of a shortest edge. Under this assumption, they gave a PTAS using a so-called partition and shifting strategy that we will also employ.

Our approach is also based on the ideas used by Fan, Zhang and Wang [3] who presented a PTAS for the minimum weight connected vertex cover problem in unit disk graphs, provided the problem satisfies a c-local condition. Inspired by their work, we define an instance of MWCVCP3 to be c-local (for some positive constant c) if in a solution S obtained by our constant-factor approximation algorithm (to be presented later), the maximum weight of the vertices in S is at most c; here we assume without loss of generality that every vertex has weight at least one. As far as we know, there is no constant-factor approximation algorithm known for the minimum weight connected vertex cover problem in unit disk graphs without any additional restriction. It has been proved by Fujito [4] that the latter problem for general graphs can be approximated within a factor of $\ln n + 3$, but not within $(1-\varepsilon)\ln n$ for any $\varepsilon > 0$ unless NP $\subset DTITM(n^{O(\log\log n)})$. So, in particular this problem is NP-hard in general.

However, as far as we know it has not been proved before that MWCVCP3 is NP-hard.

1.4. Organisation of the paper

In Section 2, we start with some additional definitions and notation, and present some preliminary results, including the NP-hardness proof for MWCVCP3 restricted to grid graphs, a subclass of unit disk graphs. In Section 3, we present our PTAS for MWCVCP3 on unit disk graphs, which is based on the partition and shifting strategy. The correctness proof, an analysis of the time complexity, and an analysis of the performance ratio are given in Section 4. Finally, we give some concluding remarks in Section 5.

2. Preliminaries

In this section, we first introduce some useful definitions and notations that will be used in the partition and shifting strategy. However, for convenience, we start by giving a very rough outline of our approach.

2.1. An outline of the approach

The overall design of our algorithm is as follows. Firstly, we determine a square area that is large enough to make sure that all the disks are contained in it. We partition this area into smaller squares. For each small square e, we define an inner area I_e and a boundary area B_e. Secondly, we adopt a constant-factor approximation algorithm to obtain a solution C_0 for MWCVCP3 restricted to the large square, such that the weight of each vertex in it is at most c, where c is a constant. Then, for each inner area I_e of every small square, we enumerate all vertex cover P_3 sets, and only consider those satisfying a certain neighborhood condition mentioned in Phase 2 of our algorithm, and choose one with a minimum total weight. Call such a vertex cover P_3 set a local optimal solution for that particular square. By the choice of the boundary areas B_e, we can ensure that the output of our algorithm is a connected vertex cover P_3 set. For this, we add all those vertices of C_0 that are located in the boundary areas of the small squares of the partition to the union of the local optimal solutions of all the squares. By adopting a shifting strategy we are able to choose a partition such that the total weight of the vertices of C_0 lying in these boundary areas of this partition is small enough.

For a given geometric representation of a connected unit disk graph $G=(V,E)$ with $|V|=n$, we initially find a minimal square Q that contains all the disks related to G. Without loss of generality, assume $Q=\{(x,y)\mid 0\le x\le q, 0\le y\le q\}$, where q depends on n. Let m be a (large) fixed integer that will be specified later. Using the partition strategy, we divide Q into smaller $m\times m$ squares. Set $p=\lfloor\frac{q}{m}\rfloor+1$. We now use a shifting strategy, as follows. We first enlarge the area of the square Q to $\widetilde{Q}=\{(x,y)\mid -m\le x\le pm, -m\le y\le pm\}$. Define this partition as $P(0)$, and denote by $P(a)$ the partition obtained from $P(0)$ by shifting the left-bottom corner of $P(0)$ from $(-m,-m)$ to $(-m+a,-m+a)$, where a is an integer and $0\le a\le m-1$ (see Fig. 1).

For each square e, we define the *inner* area I_e and the *boundary* area B_e as follows. Assume

$$e=\big\{(x,y)\;\big|\; im\le x\le (i+1)m,\ jm\le y\le (j+1)m\big\}.$$

Define

$$I_e=\big\{(x,y)\;\big|\; im+1\le x\le (i+1)m-1,\ jm+1\le y\le (j+1)m-1\big\},$$

$$B_e=e-\big\{(x,y)\;\big|\; im+3\le x\le (i+1)m-3,\ jm+3\le y\le (j+1)m-3\big\}.$$

Note that I_e and B_e have an overlap of width 2 (see Fig. 2). This will ensure that the output of our algorithm is a connected vertex cover P_3 set.

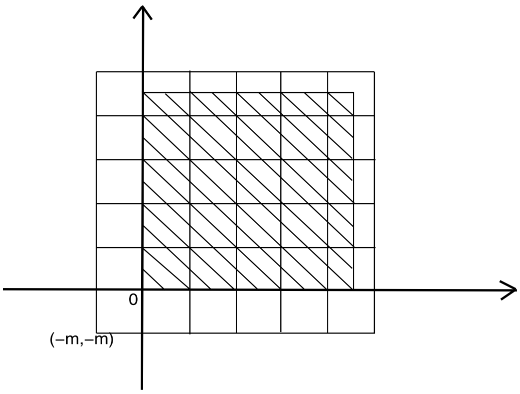

Fig. 1. Square $\widetilde{Q}$ and partition $P(0)$.

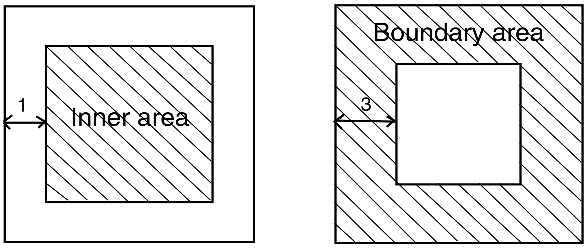

Fig. 2. Inner area and boundary area have an overlap of width 2.

In the following, we first prove the NP-hardness of MWCVCP3 restricted to grid graphs, so as a corollary we obtain the NP-hardness for unit disk graphs, what we need for our paper.

2.2. NP-hardness and approximation

Firstly, we introduce a crucial lemma due to Valiant [12] and some preliminaries for proving the complexity result. After the complexity proof, we present the constant-factor approximation result that will be used in our PTAS algorithm.

Lemma 1. *(See [12].) A planar graph $G=(V,E)$ with maximum degree 4 can be embedded in the plane using $O(|V|)$ area in such a way that its vertices are at integer coordinates (x,y) and its edges are drawn so that they are made up of a number of line segments of the form $x=i$ (we refer to this as horizontal line segments) or $y=j$ (vertical line segments), for integers i and j.*

The above lemma plays an important role in the construction of a grid graph G' corresponding to the planar graph instance G of the following decision version of the minimum connected vertex cover problem for planar graphs with maximum degree 4.

Instance: Given a planar graph $G=(V,E)$ with maximum degree 4 and a positive integer k.
Question: Does G have a connected vertex cover with at most k vertices?

The decision version of the minimum weighted connected vertex cover P_3 problem for grid graphs is stated as follows.

Instance: Given a grid graph $G'=(V',E')$, a positive integer k' and a nonnegative weight for every vertex $v' \in V'$.
Question: Does G' have a connected vertex cover P_3 set with weight at most k'?

We now have all the ingredients to present and prove our complexity result.

Theorem 1. *The decision version of the minimum weighted connected vertex cover P_3 problem is NP-complete for grid graphs.*

Proof. It is not hard to see that this decision problem is in NP. To complete the complexity proof we use a reduction from the minimum connected vertex cover problem in planar graphs with maximum degree 4, which was shown to be NP-complete in [6]. We transform a planar graph $G=(V,E)$ with maximum degree 4 into a grid graph G' such that G has a connected vertex cover S with $|S| \le k$ if and only if G' has a connected vertex cover P_3 set S' with weight at most k'. We assume without loss of generality that G is connected.

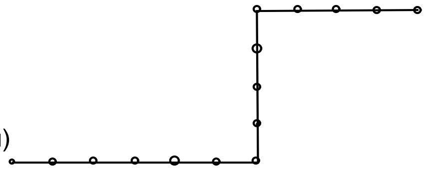

Fig. 3. A possible subgraph of G' corresponding to an edge uv of G.

Using Lemma 1, we first embed G in a 2-dimensional grid with edges represented by horizontal and vertical line segments of length at least four, and with parallel lines at least a grid square apart. The set V' of vertices of our grid graph G' will be made up of two sets. The first set, which we denote as V_1, consists of all the grid points corresponding to the vertices of G; the second set, denoted as V_2, consists of all the grid points that are internal to the paths corresponding to the line segments that represent the edges of G. Note that all these paths representing the edges of G have at least one internal P_3 (i.e., not containing a vertex corresponding to the original graph G). For each vertex $u \in V$, we denote the vertex of V_1 corresponding to u by $f(u)$. For each edge $uv \in E$, we denote the set of vertices of V_2 in the path corresponding to uv by $f\{u,v\}$. Fig. 3 shows what a subgraph of G' corresponding to an edge uv of G might look like. Now, we define a weight function $w: V' \to \{1, 2, \ldots\}$ by $w(f(v)) = |V_2|$ for every $v \in V$, and $w(v) = 1$ for every $v \in V_2$.

The construction of G' can clearly be accomplished in polynomial time. To complete our proof, we claim that there exists a connected vertex cover S in G with $|S| \le k$ if and only if there exists a connected vertex cover P_3 set S' in G' with weight at most $(k+1)|V_2|$.

First suppose that the desired connected vertex cover S exists in G. If we take S' to be the union of $f(S) = \{f(v) \mid v \in S\}$ and V_2, then, since S is a connected vertex cover in G, the subgraph induced by $f(S)$ and V_2 in G' is connected. Moreover, all the neighbors of a vertex $f(u)$ for $u \in V \setminus S$ are in V_2, and hence in S', so S' is indeed a connected vertex cover P_3 set in G', with $w(S')$ at most $|S||V_2| + |V_2| \le (k+1)|V_2|$.

For the converse, suppose that there is a connected vertex cover P_3 set S' with $w(S')$ at most $(k+1)|V_2|$ in G'. We are going to show that G has a connected vertex cover of cardinality at most k. We may assume that E is not empty; otherwise there is nothing to prove. Since E is not empty, there is at least one path between vertices of $f(V)$ in G', so with an internal P_3 that is covered by S'. So, S' contains at least one vertex of V_2. Since $w(S') \le (k+1)|V_2|$, this implies that S' contains at most k vertices of $f(V)$. In the following, it remains to show that $S = \{v \in V \mid f(v) \in S'\}$ is a connected vertex cover of G. Clearly, S induces a connected subgraph of G because $f(S)$ together with paths of V_2-vertices induces a connected subgraph of G'. Suppose there is an edge $e = uv$ of E that is not covered by S. Then neither $f(u)$ nor $f(v)$ is in S. Since $G'[S']$ is connected, this implies that no V_2-vertices on the path P of G' between $f(u)$ and $f(v)$ representing $e = uv$ belong to S'. But then P contains an internal P_3 that is not covered by S', a contradiction. This completes the proof. □

The PTAS that we will present in the next section is based on the following result.

Lemma 2. *There exists a ρ-approximation algorithm (with $\rho = 5.475$) to obtain a minimum weight connected vertex cover P_3 set in a weighted unit disk graph G, if we assume that $\delta(G) \ge 2$.*

Proof. For any weighted unit disk graph G and any given vertex cover P_3 set S, denote by T_{CVCP_3} and T_{OPT} an optimum connected vertex cover P_3 set of G and (the vertex set of) an optimum Steiner tree of G on S, respectively. Since $\delta(G) \ge 2$, the induced graph $H = G[S \cup T_{CVCP_3}]$ is connected, so H contains a Steiner tree on S. Thus, $w(T_{OPT} \setminus S) \le w(T_{CVCP_3})$.

By using the primal-dual approximation algorithm of [11], we can obtain a vertex cover P_3 set S of G with $w(S) \le 2w(T^*) \le 2w(T_{CVCP_3})$, where T^* is an optimum vertex cover P_3 set of G. Then, using the approximation algorithm for node-weighted Steiner tree on unit disk graphs of [15] on S, we can obtain a Steiner tree T with $w(T \setminus S) \le 2.5\sigma w(T_{OPT} \setminus S)$, where $\sigma = 1.39$ is the best approximation ratio for the classical Steiner tree problem [2] so far. Clearly, $V(T)$ is a connected vertex cover P_3 set of G, so we get

$$\begin{aligned} w\big(V(T)\big) &= w(S) + w\big(V(T) \setminus S\big) \\ &\le 2w(T_{CVCP_3}) + 2.5\sigma w\big(V(T_{OPT}) \setminus S\big) \\ &\le 5.475 w(T_{CVCP_3}). \qquad \square \end{aligned}$$

3. The algorithm for the PTAS

We describe the algorithm in this section. We denote the boundary area of a partition $P(a)$ as $B(P(a)) = \bigcup_{e \in P(a)} B_e$. The algorithm is executed in two phases.

Phase 1 Using the geometric representation of the graph G (with $\delta(G) \ge 2$), adopt a ρ-approximation algorithm to obtain a minimum weight connected vertex cover P_3 set C_0, where ρ is a constant (say $\rho = 5.475$, by applying the algorithm of Lemma 2). We assume that each vertex $u \in C_0$ satisfies the c-local requirement: $w(u) \le c$, where c is a fixed positive number.

Denote by $C_0(a) = C_0 \cap B(P(a))$ the set of vertices of C_0 lying in the boundary area of partition $P(a)$. Use the shifting strategy to select a partition $P(a^*)$ such that $w(C_0(a^*)) = \min\{w(C_0(a)) \mid 1 \le a \le m-1\}$.

Phase 2 For every square $e \in P(a^*)$, denote by G_e the subgraph of G induced by the vertices in I_e. Consider all the vertex cover P_3 sets C_{I_e} of G_e that satisfy the following condition:

For each component H of $(C_0 \cap (I_e \setminus B_e)) \setminus C_{I_e}$, the vertices of $N(H)$ are in the same component of $G[C_{I_e}]$, where $N(H)$ is the set of vertices adjacent to a vertex of H in the square e. $(*)$

Among all such vertex cover P_3 sets, we use exhaustive search to find a minimum weight vertex cover P_3 set $C^*_{I_e}$ of G_e.

Final Output $C = C_0(a^*) \cup (\bigcup_{e \in P(a^*)} C^*_{I_e})$.

4. Analysis of the algorithm

In this section, we analyze our algorithm on three aspects. Firstly, we prove the correctness of the algorithm. Then, we analyze the time complexity, pointing out that the algorithm can be executed in polynomial time. Finally, we prove the performance ratio of the algorithm, which turns out to be $(1+\varepsilon)$ for some arbitrarily small positive constant ε.

4.1. Correctness

In this subsection, we prove that the output C of our algorithm is a connected vertex cover P_3 set for the graph $G = (V, E)$. Firstly, we prove that C is a vertex cover P_3 set for G, and then we prove that the induced subgraph $G[C]$ is connected.

Lemma 3. *C is a vertex cover P_3 set for $G = (V, E)$.*

Proof. Let (u, v, w) be any path with length 2. Firstly, suppose (u, v, w) lies completely in one square e. Since I_e and B_e have an overlap of width 2, and the Euclidean distance between u and v, and between v and w is no more than one, there are two cases to be considered. The first case is that u, v and w belong to the inner area I_e. Because $C^*_{I_e}$ is a vertex cover P_3 set of G_e and (u, v, w) is a path with length 2 in G_e, at least one of u, v and w is in $C^*_{I_e}$. The second case is that u, v and w belong to the boundary area B_e. According to Phase 1 of the algorithm, C_0 is a minimum weight connected vertex cover P_3 set of G. So at least one of u, v and w is in C_0, and thus at least one of u, v and w is in $C_0(a^*)$. In any case, the path (u, v, w) is covered by $C^*_{I_e} \cup C_0(a^*) \subseteq C$. The case that (u, v, w) crosses two adjacent squares can be considered similarly to the second case. We conclude that C is a vertex cover P_3 set for $G = (V, E)$. □

Lemma 4. *The induced subgraph $G[C]$ is connected.*

Proof. Since $G[C_0]$ is connected and $\delta(G) \ge 2$, the graph induced by $C_0 \cup C$ is also connected. We will first show that for all squares e, removing all vertices in $(C_0 \cap (I_e \setminus B_e)) \setminus C^*_{I_e}$ from $C_0 \cup C$ results in a connected vertex cover P_3 set of G. For this purpose, let R be a component of $(C_0 \cap (I_e \setminus B_e)) \setminus C^*_{I_e}$. From condition $(*)$ in Phase 2 of our algorithm, we see that R has all its neighbors belonging to the same component of $G[C^*_{I_e}]$. Hence every P_3 extending to an edge incident with R is covered by $C^*_{I_e}$, and deleting the vertices of $V(R)$ does not disconnect the subgraph, i.e., the subgraph induced by $(C_0 \cup C) \setminus V(R)$ is still connected. Therefore, the subgraph induced by $(C_0 \cup C) \setminus (\bigcup_{e \in P(a^*)} (C_0 \cap (I_e \setminus B_e)) \setminus C^*_{I_e})$ is connected. Noting that $(C_0 \cup C) \setminus (\bigcup_{e \in P(a^*)} (C_0 \cap (I_e \setminus B_e)) \setminus C^*_{I_e}) \subseteq C_0(a^*) \cup (\bigcup_{e \in P(a^*)} C^*_{I_e}) = C$, it is not difficult to see that $G[C]$ is connected. □

Based on Lemmas 3 and 4, we obtain the following theorem, which proves the correctness of our algorithm.

Theorem 2. *The output C of our algorithm is a connected vertex cover P_3 set for the graph G.*

4.2. Time complexity

In this section, we show that our algorithm runs in polynomial time. Phase 1 of our algorithm can be executed in polynomial time to obtain a ρ-approximation solution. Phase 2 uses exhaustive search to achieve the desired solution. This is the most time consuming part, so we need to prove that this part can also be completed within polynomial time. In the following, we consider the time complexity of the algorithm for the more general case of the minimum connected vertex cover P_k problem. We use the following two results on this more general case that are copied without proofs from [8].

Lemma 5. *Let H be a connected subgraph induced by some vertices in a square e in $P(a^*)$. Assume that H does not contain a P_k. Let Δ be the maximum degree of vertices in H. Then $\Delta \le 6k$.*

Lemma 6. *For any $m \times m$ square e of the partition $P(a)$, let H be a connected subgraph that does not contain a P_k in a square e. Then $|V(H)| \le g(k)$, where $g(k) = 4(6k)^k$.*

We also need the following lemma.

Lemma 7. *The number of independent unit disks in an $m \times m$ square e is at most $\lceil \frac{4(m+2)^2}{\pi} \rceil$.*

Proof. Enlarge the $m \times m$ square to an $(m+2) \times (m+2)$ square by adding a boundary with width one. Then all the disks whose central points are in the $m \times m$ square lie completely in the $(m+2) \times (m+2)$ square. Since each unit disk occupies an area of $\pi/4$, the result follows from the independence assumption. □

Theorem 3. *The running time of our algorithm is $n^{O(1/\varepsilon^2)}$, where n is the number of vertices of the graph (and ε is a constant depending on m and c).*

Proof. For every square $e \in P(a^*)$, let G_e be the subgraph of G induced by the vertices in I_e, and let $Comp(G_e)$ be the set of components of G_e. For any component $H \in Comp(G_e)$, let C_H be a vertex cover P_3 set of H, and consider the subgraph $G[V(H) \setminus C_H]$ induced by $V(H) \setminus C_H$. We know that $G[V(H) \setminus C_H]$ does not contain any P_k. Without loss of generality, we may assume that $V(H) \setminus C_H = C_1 \cup C_2 \cup \dots \cup C_s$, where C_i $(i = 1, 2, \dots, s)$ are components of $G[V(H) \setminus C_H]$ and s is at most the maximum number of independent unit disks in the square e. By Lemma 7, we have

$$s \le \left\lceil \frac{4(m+2)^2}{\pi} \right\rceil.$$

According to Lemma 6, we have

$$\left|V(H) \setminus C_H\right| = |C_1 \cup C_2 \cup \dots \cup C_s| \le g(k) \left\lceil \frac{4(m+2)^2}{\pi} \right\rceil.$$

Now we use the following strategy to compute C_H: enumerate the induced subgraphs of H with no more than $g(k)\lceil \frac{4(m+2)^2}{\pi} \rceil$ vertices to find all induced subgraphs whose components do not contain a P_k. Then take complements and find the one which is of minimum total weight satisfying condition (∗).

The above exhaustive search for C_H takes time at most

$$\sum_{i=0}^{g(k)\lceil \frac{4(m+2)^2}{\pi} \rceil} \binom{n_H}{i} = n_H^{O(g(k)m^2)},$$

where n_H is the number of vertices of H. Therefore, the total running time of our algorithm is at most

$$\sum_{e,H} n_H^{O(g(k)m^2)} = \left(\sum_{e,H} n_H\right)^{O(g(k)m^2)} = n^{O(g(k)m^2)} = n^{O(g(k)/\varepsilon^2)}.$$

(As we will see at the end of the proof of Theorem 4, $m = \lceil \frac{(12+120c)\rho}{\varepsilon} \rceil$ is a suitable choice for our purposes.) So in case $k = 3$, we conclude that the running time of our algorithm is $n^{O(1/\varepsilon^2)}$. □

4.3. Performance analysis

In this section, we prove that our algorithm has performance ratio $(1+\varepsilon)$ for arbitrarily small $\varepsilon > 0$. Firstly, we introduce some definitions and lemmas that will be used in this section.

Definition 1. For two subgraphs G_1 and G_2 of a (connected) graph G, denote by $dist(G_1, G_2)$ the distance between G_1 and G_2, which is the length of a shortest path of G connecting G_1 and G_2. If $dist(G_1, G_2) = k$, we also say that there are $k-1$ vertices connecting G_1 and G_2.

The following lemma shows that for any vertex cover P_3 set C of a connected graph H that does not induce a connected subgraph, there always exist two components of $H[C]$ at distance two or three.

Lemma 8. *Suppose H is a connected graph, and let C be a vertex cover P_3 set of H. If the subgraph $H[C]$ is not connected, then there exist two components R_1 and R_2 in $H[C]$ such that $dist(R_1, R_2) = 2$ or 3.*

Proof. Since C is a vertex cover P_3 set of H, the graph $G[V-C]$ only consists of isolated vertices and isolated edges. If all components of $G[C]$ are mutually at distance at least 4, then there clearly exists a P_3 in $G[V-C]$, a contradiction. □

The following well-known folklore property of unit disk graphs plays an important role in the approximation analysis.

Lemma 9. *Suppose G is a unit disk graph, and let $u \in V(G)$. Then there are at most 5 independent vertices in $N(u)$, where $N(u)$ is the set of vertices adjacent to u in G.*

Based on the above definition and lemmas, we prove that our algorithm is a PTAS.

Theorem 4. *Suppose C^* is a minimum weight connected vertex cover P_3 set of G, and suppose that C is the output of our algorithm. Then $w(C) \le (1+\varepsilon)w(C^*)$.*

Proof. We prove the statement in two steps.

Firstly, we study the weight of the vertices in the boundary area, and show that

$$w\big(C_0(a^*)\big) \le \frac{12\rho}{m} w\big(C^*\big), \tag{4.1}$$

where m only depends on ε and c.

When we adopt the shifting strategy, it can be observed that a vertex of C_0 appears at most 12 times in the boundary area of $B(P(a))$s. Therefore, we have

$$w\big(C_0(0)\big) + w\big(C_0(1)\big) + \cdots + w\big(C_0(m-1)\big) \le 12w(C_0).$$

Combining this with $w(C_0) \le \rho w(C^*)$, we have

$$w\big(C_0(a^*)\big) \le \frac{12\rho}{m} w\big(C^*\big).$$

Next, we analyze the weight $w(\bigcup_{e\in P(a^*)} C^*_{I_e})$ of the vertices chosen in the inner area I_e. For each square e, let $C^*_e = C^* \cap I_e$. In order to compare the weight of the optimal solution C^* with $w(\bigcup_{e\in P(a^*)} C^*_{I_e})$, we need to modify C^*_e into a feasible local solution for each square e, i.e., C^*_e must satisfy requirement $(*)$ in Phase 2. If this is not the case already, then there must exist a component R in $(C_0 \cap (I_e \setminus B_e)) \setminus C^*_e$ whose neighbors are in at least two different components of $G[C^*_e]$; then, using Lemma 8, we add at most two vertices of $V(R)$ to C^*_e in order to reduce the number of components of $G[C^*_e]$. If the new C^*_e still does not satisfy requirement $(*)$, we continue as above by adding vertices in order to merge components. Notice that $R \subseteq C_0$, so the weight of the added vertices of $V(R)$ is at most c. Suppose we have to do this k times in order to get a new set $\widetilde{C^*_e}$ satisfying $(*)$. Then

$$w\big(\widetilde{C^*_e}\big) \le w\big(C^*_e\big) + 2kc. \tag{4.2}$$

On the other hand, we can prove that

$$\big|C_0\big(a^*\big) \cap e\big| \ge \frac{k}{5}. \tag{4.3}$$

For this purpose, we assume that the components merged are in the order that firstly R_1 is merged with R_2, then R_3 with $R_4, \ldots,$ and finally R_{2m-1} with R_{2m}.

Firstly, suppose these components are all different. For each $i = 1, 2, \ldots, k$, let u_i, v_i be the two vertices lying in $V(R_{2i-1}) \cap B_e \cap I_e$, and $u_i v_i$ be an edge of G, such that u_i is adjacent with a vertex $w_i \in B_e \backslash I_e$. Such u_i, v_i exist since R_{2i-1} is connected to the outer parts of e through C^*. Since $w_i u_i v_i$ is a P_3, then at least one of them belongs to C_0, denoted by z_i. Note that $w_i, u_i, v_i \in B_e$, hence $z_i \in C_0(a^*) \cap e$. A vertex may be used more than once as z_i. For instance, there may be two indices $i \ne j$ such that the vertex of C_0 is covering the two P_3s $w_i u_i v_i$ and $w_j u_j v_j$, and at the same time $w_i = w_j$. Since R_i and R_j are two distinct components in $G[C^* \cap I_e]$, we see that $u_i v_i$ and $u_j v_j$ are two independent edges. Then it follows from Lemma 9 that the number of times such a vertex serves as z_i is at most 5. Hence inequality (4.3) holds.

Next, consider the case that these components are not all distinct. For example, assume component R_3 is obtained through merging R_1 and R_2. Then $u_3 v_3$ can be chosen from $V(R_2) \cap B_e \cap I_e$, which is independent from $u_1 v_1$. Generally, we can find k independent edges $u_1 v_1, u_3 v_3, \ldots, u_{2m-1} v_{2m-1}$, so inequality (4.3) also holds.

According to inequalities (4.2) and (4.3), and the assumption that the weight of any vertex is at least one, we have

$$w\big(\widetilde{C^*_e}\big) \le w\big(C^*_e\big) + 10c\big|C_0\big(a^*\big) \cap e\big|. \tag{4.4}$$

Since in Phase 2 of our algorithm, $C^*_{I_e}$ is a minimum weight vertex cover P_3 set satisfying condition $(*)$, we have $w(C^*_{I_e}) \le w(\widetilde{C^*_e})$.

Combining this with inequalities (4.1) and (4.4), we have

$$\begin{aligned} w(C) &\le w\big(C_0(a^*)\big) + \sum_{e\in P(a^*)} \big(w\big(C^*_{I_e}\big)\big) \\ &\le w\big(C_0(a^*)\big) + \sum_{e\in P(a^*)} \Big(w\big(C^*_e\big) + 10c\big|C_0(a^*)\cap e\big|\Big) \\ &\le w\big(C_0(a^*)\big) + w\big(C^*\big) + 10c\big|C_0(a^*)\big| \\ &\le w\big(C_0(a^*)\big) + w\big(C^*\big) + 10cw\big(C_0(a^*)\big) \\ &\le \left(1 + \frac{(12+120c)\rho}{m}\right) w\big(C^*\big). \end{aligned}$$

So if we let $m = \lceil \frac{(12+120c)\rho}{\varepsilon} \rceil$ in our algorithm, then $w(C) \le (1+\varepsilon)w(C^*)$. This completes the proof of the theorem. □

5. Conclusion

We showed that the minimum weight connected vertex cover P_3 problem is NP-hard for grid graphs, a subclass of unit disk graphs. Moreover, we presented a polynomial time approximation scheme for this problem for unit disk graphs with a given geometric representation, under the additional conditions that the problem is c-local and that the unit disk graph has minimum degree at least 2. We strongly believe that a completely different approach is needed in case the c-local condition is dropped, but we think our approach can be modified for the case vertices with degree 1 are allowed, but it will get more tedious.

Acknowledgements

The authors would like to thank Dr. Jianhua Tu and Dr. Wenli Zhou for introducing the vertex cover P_3 problem to them and for providing useful background information. Part of the work was done while the second author visited the University of Twente. Support from the University of Twente is gratefully acknowledged. The authors also thank the anonymous referees for useful comments that improved the presentation.

References

[1] J.A. Bondy, U.S.R. Murty, Graph Theory, Grad. Texts in Math., vol. 244, Springer, New York, 2008.
[2] J. Byrka, F. Grandoni, T. Rothvoss, L. Sanita, An improved LP-based approximation for Steiner tree, in: Proc. of 42th STOC, 2010, pp. 583–592.
[3] L.D. Fan, Z. Zhang, W. Wang, PTAS for minimum weighted connected vertex cover problem with c-local condition in unit disk graphs, J. Comb. Optim. 22 (2011) 663–673.
[4] T. Fujito, On approximability of the independent/connected edge dominating set problems, Inform. Process. Lett. 79 (2001) 261–266.
[5] M.R. Garey, D.S. Johnson, Computers and Intractability, W.H. Freeman and Company, San Francisco, 1979.
[6] M.R. Garey, D.S. Johnson, The rectilinear Steiner tree is NP-complete, SIAM J. Appl. Math. 32 (1977) 826–834.
[7] J. Kratochvíl, Intersection graphs of noncrossing arc-connected sets in the plane, in: Proc. Symposium on Graph Drawing, in: Lecture Notes in Comput. Sci., vol. 1190, 1997, pp. 257–270.
[8] X.L. Liu, H.L. Lu, W. Wang, W.L. Wu, PTAS for the minimum k-path connected vertex cover problem in unit disk graphs, J. Global Optim. 56 (2013) 449–458.
[9] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, Boca Raton, 1996.
[10] M. Novotny, Design and analysis of a generalized Canvas protocol, in: Proceedings of WISTP, in: Lecture Notes in Comput. Sci., vol. 6033, 2010, pp. 106–121.
[11] J.H. Tu, W.L. Zhou, A primal-dual approximation algorithm for the vertex cover P_3 problem, Theoret. Comput. Sci. 412 (2011) 7044–7048.
[12] L.G. Valiant, Universality considerations in VLSI circuits, IEEE Trans. Comput. 30 (1981) 135–140.
[13] L.S. Wang, T. Jiang, An approximation scheme for some steiner tree problems in the plane, Networks 28 (1996) 187–193.
[14] Z. Zhang, X.F. Gao, W.L. Wu, PTAS for connected vertex cover in unit disk graphs, Theoret. Comput. Sci. 410 (2009) 5398–5402.
[15] F. Zou, X.Y. Li, S.G. Gao, W.L. Wu, Node-weighted Steiner tree approximation in unit disk graphs, J. Global Optim. (2009) 342–349.