RAIRO-Theor. Inf. Appl. 45 (2011) 331–346
DOI: 10.1051/ita/2011106

Available online at:
www.rairo-ita.org

A NOTE ON MAXIMUM INDEPENDENT SETS AND MINIMUM CLIQUE PARTITIONS IN UNIT DISK GRAPHS AND PENNY GRAPHS: COMPLEXITY AND APPROXIMATION

Marcia R. Cerioli[1], Luerbio Faria[2], Talita O. Ferreira[3] and Fábio Protti[4]

Abstract. A *unit disk graph* is the intersection graph of a family of unit disks in the plane. If the disks do not overlap, it is also a *unit coin graph* or *penny graph*. It is known that finding a maximum independent set in a unit disk graph is a NP-hard problem. In this work we extend this result to penny graphs. Furthermore, we prove that finding a minimum clique partition in a penny graph is also NP-hard, and present two linear-time approximation algorithms for the computation of clique partitions: a 3-approximation algorithm for unit disk graphs and a 2-approximation algorithm for penny graphs.

Mathematics Subject Classification. 05C69, 05C75, 68W25, 68Q25.

Introduction

Given a family $\mathscr{F}$ of objects, the *intersection graph* of $\mathscr{F}$ is the graph whose vertices are in a one-to-one correspondence with the objects of $\mathscr{F}$ in such a way that there exists an edge joining two vertices if and only if the corresponding objects intersect. A *unit disk* is a disk of diameter one in the euclidean plane. Two unit disks *intersect* if the distance between their centers is less than or equal to one. Two unit disks *overlap* if the distance between their centers is strictly less

Keywords and phrases. Unit disk graphs, unit coin graphs, penny graphs, independent set, clique partition, approximation algorithms.

[1] Instituto de Matemática and COPPE-Sistemas, Universidade Federal do Rio de Janeiro, Brazil. Partially supported by CNPq and FAPERJ. cerioli@cos.ufrj.br

[2] FFP, Universidade do Estado do Rio de Janeiro, Brazil. Partially supported by CNPq. luerbio@cos.ufrj.br

[3] COPPE-Sistemas, Universidade Federal do Rio de Janeiro, Brazil. Supported by CNPq. talita@cos.ufrj.br

[4] Instituto de Computação, Universidade Federal Fluminense, Brazil. Partially supported by CNPq and FAPERJ. fabio@ic.uff.br

than one. A graph G is a *unit disk graph* if it is the intersection graph of a family of unit disks. When the disks do not overlap, G is also a *unit coin graph*, or *penny graph*. A *realization* of a unit disk graph (resp. penny graph) G is a family $\mathscr{F}$ of unit disks (resp. non-overlapping unit disks) such that G is the intersection graph of $\mathscr{F}$.

Intersection graphs of geometric objects have received much attention since the 70's. In this context, unit disk graphs and penny graphs appear in the modeling of several problems [3,4,6,8,16,18–20], especially with the appearance of wireless sensor networks. In a natural model for such networks, sensor nodes are vertices of a graph, and two vertices are adjacent if and only if the sensing areas of the corresponding sensor nodes intersect, which implies that they are able to exchange messages. Typically, sensors are identical, leading to a unit disk graph model. Clustering is an important aspect in many applications for wireless sensor networks. As explained in [20], sensor nodes must be grouped by some similarity or proximity criteria, so that the number of groups (clusters) is minimized. Minimum clique partition is a kind of clustering where sensor nodes are grouped to form cliques, where each clique indicates mutual proximity.

In [6], Clark *et al.* proved that the independent set problem is NP-complete when restricted to unit disk graphs. This intractability result motivated the search for approximation algorithms, particularly for PTASs (polynomial-time approximation schemes). In [18], Marathe *et al.* point out that given a unit disk graph G, the size of a maximum independent set in $G[N(v)]$ (the subgraph of G induced by the set of neighbors of v) is at most 3 if, in some realization of G, v is associated to a disk whose center has the smallest x-coordinate. This fact is used to design a simple greedy approximation algorithm which constructs an independent set S as follows: repeatedly find a vertex v with the property described above, add v to S and remove $v \cup N(v)$ from G. This algorithm has an approximation ratio of 3 and runs in $O(n^6)$ time. Since recognizing unit disk graphs is NP-Hard [4] and the complexity of constructing a realization of a unit disk graph is still an open question, most algorithms demand a realization as input. The algorithm by Marathe *et al.* does not, but if a realization of the input graph is given, its complexity decreases to $O(n^2)$.

In [16], Hunt III *et al.* present an approximation scheme for finding independent sets in unit disk graphs. The algorithm takes a realization as input, and produces a solution with size at least $(\frac{k}{k+1})^2$ times the optimal size, where k is the smallest integer such that $(\frac{k}{k+1})^2 \geq 1 - \varepsilon$, for a given $\varepsilon > 0$. The total running time of the algorithm is $n^{O(k^2)}$. The adopted strategy consists basically of generating independent sets for several "representative" subgraphs ("shifting strategy" [1]), and taking the best solution over all the subgraphs. In each subgraph, the solution is the union of several exact partial solutions.

In [20], Pirwani and Salavatipour describe a robust polynomial time approximation scheme for the computation of clique partitions in unit disk graphs. If the input graph G (with edge-lengths) is a unit disk graph, their algorithm returns a clique partition of G with size $1+\varepsilon$ times the optimum; otherwise, it either outputs a clique partition with no performance guarantee or detects that G is not a unit disk graph. In the same work, the authors describe a $(2+\varepsilon)$-approximation algorithm for a weighted version of the clique partition problem in unit disk graphs, assuming that they are given in standard form (no realization is provided); this result improves the previous 8-approximation ratio described in [21] for the unweighted clique partition problem in unit disk graphs (given in standard form).

Typically, for a desired $\varepsilon > 0$, the running time of a polynomial time approximation scheme depends upon a factor $n^{f(1/\varepsilon)}$. An alternative for high polynomial running times is to devise simple approximation algorithms with low time complexity. This is what we develop in Section 3. Of course, any approximation algorithm designed for unit disk graphs maintains its performance when applied to a penny graph. One interesting, general question is thus to find examples of problems for which the more restricted structure and additional theoretical properties of penny graphs may lead to algorithmic improvements. We show in Section 3 that the clique partition problem fits into this context.

Other studies on penny graphs are described in the sequel. The sum coloring problem, which consists of finding a proper coloring that minimizes the sum of the colors (positive integers) over all vertices, is NP-hard for penny graphs [5]. The edge extremal problem for penny graphs, which aims to find the maximum number $E(n)$ of edges of a penny graph on n vertices, was originally posed by Reutter [22] and Erdös [9], and solved by Harborth [15]; Harborth found the expression $E(n) = \lfloor 3n - \sqrt{12n-3} \rfloor$, which corresponds to the maximum possible number of tangency points that can be obtained when arranging n equal-sized coins on a plane surface.

We now summarize the contributions of this work. In Section 1, we extend Clark's result [6] by proving that the independent set problem remains NP-complete when restricted to the class of penny graphs. In Section 2, we also prove the NP-completeness of the clique partition problem when restricted to penny graphs, thus answering the open question of determining the complexity of the clique partition problem for unit disk graphs (see [23], p. 316). Finally, in Section 3, we present two approximation algorithms for finding clique partitions: a 3-approximation algorithm for unit disk graphs which runs in time linear in the size of the complement of the input graph, and a 2-approximation algorithm for penny graphs which runs in $O(n)$ time, where n is the number of vertices of the input graph.

An extended abstract of this work has previously appeared in [7].

Description of the problems

In the sequel, we give the formal definitions of the problems dealt with in this work.

> INDEPENDENT SET [12]
> *Instance:* A Graph $G = (V, E)$ and a positive integer $k \leq |V|$.
> *Question:* Does G contain an independent set of size at least k, *i.e.*, a subset $V' \subseteq V$ such that $|V'| \geq k$ and no two vertices in V' are joined by an edge in E?

VERTEX COVER [12]
Instance: A Graph $G = (V, E)$ and a positive integer $k \leq |V|$.
Question: Is there a vertex cover of size at most k for G, *i.e.*, a subset $V' \subseteq V$ such that $|V'| \leq k$ and for each edge $(u, v) \in E$ at least one of u, v belongs to V'?

CLIQUE PARTITION [12]
Instance: A Graph $G = (V, E)$ and a positive integer $k \leq |V|$.
Question: Is there a clique partition of size at most k for G, *i.e.*, a collection $V_1, V_2, \ldots, V_\ell$ of disjoint subsets of V such that $\ell \leq k$, $\cup_{i=1}^{\ell} V_i = V$, and each V_i is a clique? (The cliques are not necessarily maximal.)

PLANAR 3SAT [10]
Instance: A set of variables U and a collection of clauses $\mathcal{Q}$ over U such that each clause contains at most three literals and the undirected graph $G_{\mathcal{Q}} = (V, E)$ is planar, where $V = U \cup \mathcal{Q}$ and $E = \{(u_i, Q_j) \mid u_i \in Q_j \text{ or } \overline{u_i} \in Q_j\}$.
Question: Is there a truth assignment that satisfies $\mathcal{Q}$?

PLANAR 3SAT$_{\overline{3}}$
Instance: A set of variables U and a collection of clauses $\mathcal{Q}$ over U such that each clause contains at most three literals, each variable occurs at most three times in $\mathcal{Q}$, and the undirected graph $G_{\mathcal{Q}} = (V, E)$ is planar, where $V = U \cup \mathcal{Q}$ and $E = \{(u_i, Q_j) \mid u_i \in Q_j \text{ or } \overline{u_i} \in Q_j\}$.
Question: Is there a truth assignment that satisfies $\mathcal{Q}$?

1. INDEPENDENT SET RESTRICTED TO PENNY GRAPHS

Let $G = (V, E)$. Clearly, $V' \subseteq V$ is a vertex cover of G if and only if $V \backslash V'$ is an independent set of G. Thus, we first show that VERTEX COVER restricted to penny graphs is NP-complete, based on the ideas presented in [6] (Thm. 4.1). We need the following result:

Lemma 1.1. [24] *A planar graph $G = (V, E)$ with maximum degree 4 can be efficiently embedded in the plane using $O(|V|)$ area (number of unit cells) in such a way that its vertices are at integer coordinates and its edges are drawn so that they are made up of vertical and horizontal line segments.*

Theorem 1.2. VERTEX COVER *restricted to penny graphs is NP-complete.*

Proof. The problem is clearly in NP. The reduction is done from VERTEX COVER restricted to planar graphs with maximum degree 3, which was shown to be NP-complete by Garey and Johnson in [11]. From a planar graph $G = (V, E)$ with maximum degree 3, we construct a penny graph $G' = (V', E')$ such that there is a vertex cover S of G satisfying $|S| \leq k$ if and only if there is a vertex cover S' of G' satisfying $|S'| \leq k'$, where k' is defined using the following idea. Consider a drawing of G in the plane according to Lemma 1.1 (see Figs. 1a and 1b). Lemma 1.1 is precisely the tool we need to construct (a realization of) G', as follows: create a black disk for each vertex in V, and represent each edge $(x, y) \in E$ by an even path P_{xy} consisting of $4l_{xy}$ white disks, where l_{xy} is the length of an edge between the vertices x and y (see Fig. 1c). In order to achieve the value $4l_{xy}$, local displacements in the disks can be made when necessary, as shown in Figure 2. Note that the number of vertices of G' is $|V| + \sum_{(x,y)\in E} 4l_{xy}$. Define $k' = k + \sum_{(x,y)\in E} 2l_{xy}$; as we shall see, this is a suitable value to set up the reduction. To conclude the proof, let S be a vertex cover of G satisfying $|S| \le k$. Construct S' from S by considering the $|S|$ black discs corresponding to the vertices of S, plus $2l_{xy}$ alternating white disks for each edge $(x, y) \in E$. Clearly, S' is a vertex cover of G' satisfying $|S'| \le k'$. Conversely, let S' be a vertex cover of G' satisfying $|S'| \le k'$. Analyze S' by looking at the discs corresponding to its vertices. For each edge $(x, y) \in E$, note that at least $2l_{x,y}$ alternating white discs are necessary, in G', to cover the path P_{xy} linking the black discs corresponding to x and y. Hence, S' contains at least $k' - k$ vertices corresponding to white discs. In addition, for each such path P_{xy}, there must exist a disc (other than those $2l_{xy}$ white discs) covering the contact between the black disc corresponding to x (or y), and the white disc adjacent to it in P_{xy}. We can assume without loss of generality that such an additional disc is black, that is, it corresponds to vertex x (or y). This means that the remaining k vertices of S' correspond to black discs, *i.e.*, original vertices of G. Define S by taking such k vertices. It is clear that, for each edge $(x, y) \in E$, one of x, y is necessarily in S, for otherwise one of the discs corresponding to x, y would not have been covered in S', a contradiction. Therefore, S is a vertex cover of G satisfying $|S| \le k$. □

Corollary 1.3. INDEPENDENT SET *restricted to penny graphs is NP-complete.*

2. CLIQUE PARTITION RESTRICTED TO PENNY GRAPHS

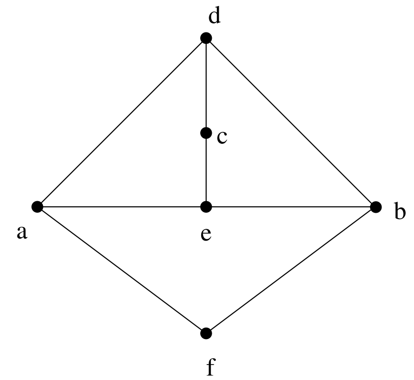
In this section we prove the NP-completeness of CLIQUE PARTITION for penny graphs. We need first the following lemma:

Lemma 2.1. PLANAR $3\text{SAT}_{\overline{3}}$ *is NP-complete.*

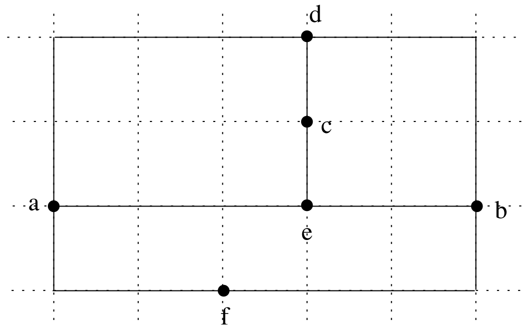
Proof. PLANAR $3\text{SAT}_{\overline{3}}$ is clearly in NP. We use the NP-completeness of PLANAR 3SAT [10] to prove that PLANAR $3\text{SAT}_{\overline{3}}$ is NP-complete.

Given an instance $I = (U, \mathcal{Q})$ of PLANAR 3SAT we define an instance $I' = (U', \mathcal{Q}')$ of PLANAR $3\text{SAT}_{\overline{3}}$ in the following way. First, set $U' = U$ and $\mathcal{Q}' = \mathcal{Q}$. Next, for each variable u in U occurring $k > 3$ times, we use an enforcement strategy (see [2] and [17]) in order to limit the number of occurrences of each variable, as follows:

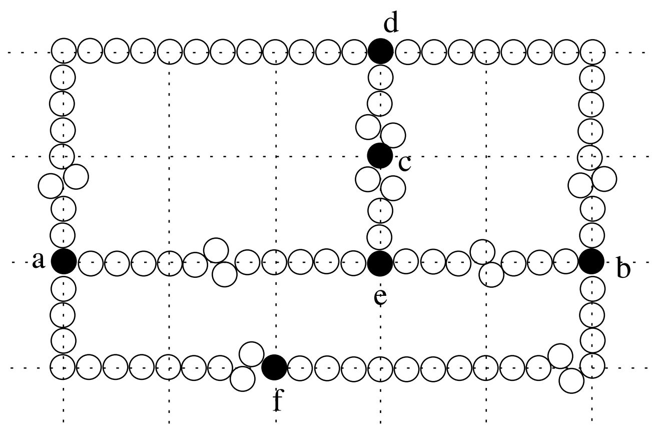
(i) remove u from U';
(ii) add new variables $u^1, u^2, \ldots, u^k$ to U';
(iii) add new clauses $(u^1 \vee \overline{u^2}), (u^2 \vee \overline{u^3}), \ldots, (u^{k-1} \vee \overline{u^k}), (u^k \vee \overline{u^1})$ to $\mathcal{Q}'$;



(a) Planar graph G with maximum degree 3



(b) Drawing of G according to Lemma 1



(c) Realization of G'

FIGURE 1. VERTEX COVER restricted to penny graphs.

(a) Odd number of disks (b) Achieving an even number of disks

FIGURE 2. Local displacements in the disks.

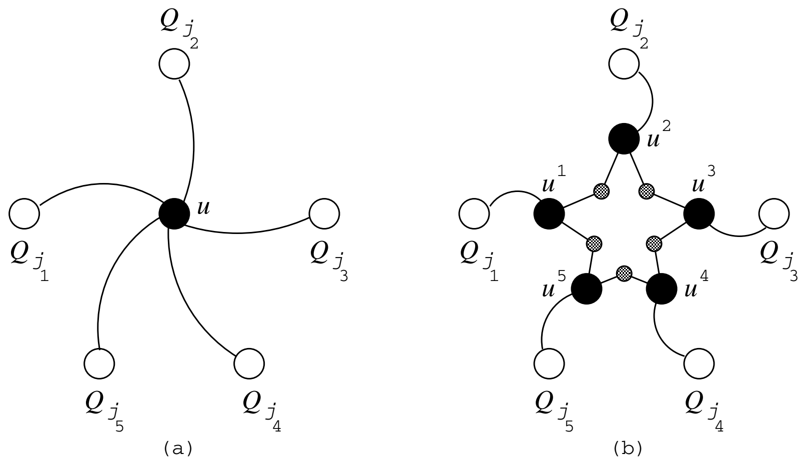


FIGURE 3. (a) Vertex u of degree $k = 5$ with neighbors $Q_{j_1}, Q_{j_2}, \ldots, Q_{j_5}$ in drawing D. (b) Vertices u^1, u^2, u^3, u^4, u^5 and clauses $(u^1 \vee \overline{u^2}), (u^2 \vee \overline{u^3}), (u^3 \vee \overline{u^4}), (u^4 \vee \overline{u^5}), (u^5 \vee \overline{u^1})$ in drawing D'.

(iv) assume that u occurs in clauses $Q_{j_1}, Q_{j_2}, \ldots, Q_{j_k}$; replace the k ocurrences of u as follows: if u occurs positively in Q_{j_i} then replace u by u^i, otherwise by $\overline{u^i}$.

In addition, let D be a plane drawing of the bipartite graph $G_{\mathcal{Q}}$ corresponding to instance $I = (U, \mathcal{Q})$, and consider the neighbors $Q_{j_1}, Q_{j_2}, \ldots, Q_{j_k}$ of u in D. See an example in Figure 3a for $k = 5$, where $Q_{j_1}, Q_{j_2}, \ldots, Q_{j_5}$ are drawn in clockwise direction around u. A plane drawing D' of the bipartite graph $G_{\mathcal{Q}'}$ corresponding to instance $I' = (U', \mathcal{Q}')$ can be obtained as shown in Figure 3b, where clauses $(u^1 \vee \overline{u^2}), (u^2 \vee \overline{u^3}), (u^3 \vee \overline{u^4}), (u^4 \vee \overline{u^5}), (u^5 \vee \overline{u^1})$ correspond to gray vertices. This concludes the construction of instance I' of PLANAR 3SAT$_{\overline{3}}$.

Now we have to prove that $I = (U, \mathcal{Q})$ is satisfiable if and only if $I' = (U', \mathcal{Q}')$ is satisfiable.

Let τ be a truth assignment for U satisfying $\mathcal{Q}$. We show how to define a truth assignment τ' for U' satisfying $\mathcal{Q}'$. If u occurs 3 times or less in $\mathcal{Q}$, set $\tau'(u) = \tau(u)$. Otherwise, if u occurs more than 3 times in $\mathcal{Q}$, let $u^1, u^2, \ldots, u^k$ be the new variables which have replaced u in U'. Note that the new clauses $(u^1 \vee \overline{u^2}), (u^2 \vee \overline{u^3}), \ldots, (u^{k-1} \vee \overline{u^k}), (u^k \vee \overline{u^1})$ in $\mathcal{Q}'$ are satisfied whenever $u^1, u^2, \ldots, u^k$ have the same truth value. Hence, set $\tau'(u^i) = \tau(u)$, $i = 1, 2, \ldots, k$. This proves that $I' = (U', \mathcal{Q}')$ is satisfiable.

Now let τ' be a truth assignment for U' satisfying $\mathcal{Q}'$. We show how to define a truth assignment τ for U satisfying $\mathcal{Q}$. If u occurs 3 times or less in $\mathcal{Q}$, set $\tau(u) = \tau'(u)$. Otherwise, if u occurs more than 3 times in $\mathcal{Q}$, note that the existence of the clauses $(u^1 \vee \overline{u^2}), (u^2 \vee \overline{u^3}), \ldots, (u^{k-1} \vee \overline{u^k}), (u^k \vee \overline{u^1})$ in $\mathcal{Q}'$ force $\tau'(u^1) = \tau'(u^2) = \cdots = \tau'(u^k)$. Hence, set $\tau(u) = \tau'(u^1)$. This proves that $I = (U, \mathcal{Q})$ is satisfiable. □

Theorem 2.2. CLIQUE PARTITION *restricted to penny graphs is NP-complete.*

Proof. The problem is clearly in NP. Given a PLANAR $3\text{SAT}_{\overline{3}}$ instance $I = (U, \mathcal{Q})$, we construct a realization of a penny graph G' from $G_{\mathcal{Q}}$ using two structures: *circuit* and *junction* (see Fig. 4). For each variable u_i we construct a *circuit* C^i of disks such that $C^i = C^i_1, C^i_2, \ldots, C^i_{r_i}, C^i_1$, where r_i is even ($r_i \geq 6$) and $C^i_k \cap C^i_\ell \neq \emptyset$ if and only if $\ell = (k \bmod r_i) + 1$ (*i.e.*, C^i_k and C^i_ℓ are consecutive in the circuit). In the remainder of this proof, assume that index ℓ always satisfies $\ell = (k \bmod r_i) + 1$. For each clause Q_j we define a *junction* T^j consisting of five disks as shown in Figure 4. Three of them, D_x, D_y and D_z, are reserved for intercepting circuits (the dotted disks in Fig. 4), as described below.

Since $G_{\mathcal{Q}}$ is a planar graph with maximum degree 3, draw $G_{\mathcal{Q}}$ in the plane using Lemma 1.1. Now, construct a realization of G' from $G_{\mathcal{Q}}$ as follows. (See an example of the construction in Figs. 5 and 6). For each Q_j, place T^j so that the center of the disk D_y has the same coordinates of the vertex of $G_{\mathcal{Q}}$ associated to Q_j. For each u_i, draw C^i in such a way that it intercepts the junctions corresponding to clauses containing u_i or $\overline{u_i}$. More precisely: if u_i occurs as a *positive* (resp. *negative*) literal in Q_j, then C^i_k and C^i_ℓ must intercept one of the disks D_x, D_y, D_z in T^j, for some *even* (resp. *odd*) integer k. Since each edge (u_i, Q_j) in the drawing of $G_{\mathcal{Q}}$ is represented by a collection of line segments on the grid, the circuits are drawn following grid segments. In order to ensure that each circuit consists of an even number of disks and the intersections involving circuits and junctions occur as described above, apply the displacements in Figure 2 when necessary. The radius of the disks is conveniently chosen to guarantee that disks belonging to distinct circuits do not intercept.

In order to cover the r_i vertices in G' corresponding to a circuit C^i, at least $\frac{r_i}{2}$ cliques are needed. The five vertices in G' corresponding to a junction T^j can be covered by two or three cliques (see Fig. 6).

Let $p = 2|\mathcal{Q}| + \sum_{i=1}^{|U|} \frac{r_i}{2}$. Let us prove that $\mathcal{Q}$ is satisfiable if and only if there is a clique partition $\mathcal{Z}$ of G' such that $|\mathcal{Z}| = p$.

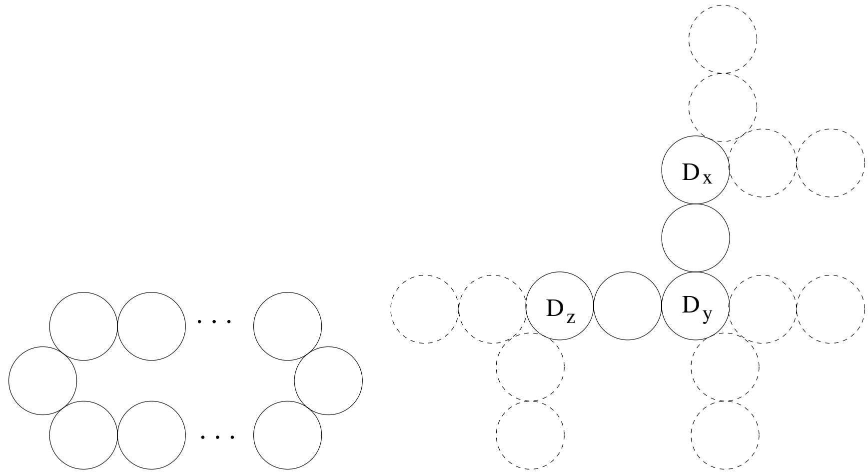


FIGURE 4. CLIQUE PARTITION restricted to penny graphs: structures of reduction.

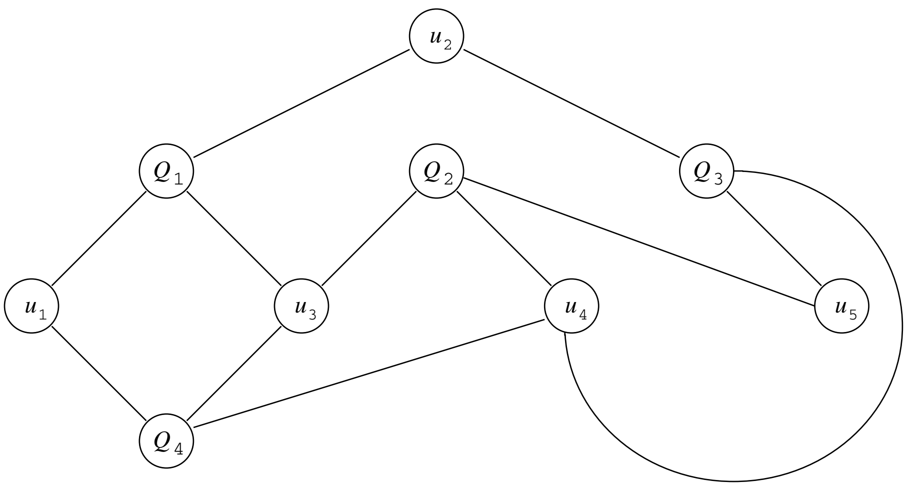
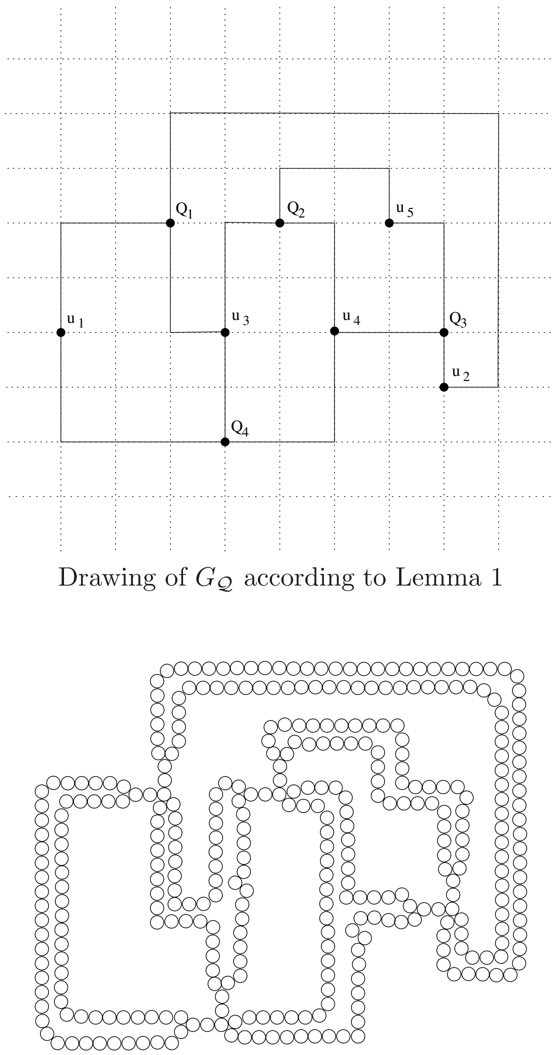


FIGURE 5. Graph $G_{\mathcal{Q}}$ constructed from $\mathcal{Q} = \{(u_1 \vee \overline{u}_2 \vee \overline{u}_3) \wedge (u_3 \vee \overline{u}_4 \vee \overline{u}_5) \wedge (u_4 \vee u_2 \vee \overline{u}_5) \wedge (\overline{u}_1 \vee u_3 \vee \overline{u}_4)\}$.

Suppose first that there exists a truth assignment τ for $\mathcal{Q}$. We construct $\mathcal{Z}$ as follows. For each circuit C^i, add to $\mathcal{Z}$ all the cliques corresponding to edges (v, w) of G' such that v corresponds to C^i_k and w corresponds to C^i_ℓ, either for



Drawing of $G_{\mathcal{Q}}$ according to Lemma 1

Realization of G'

FIGURE 6. Construction of G'.

$k = 2, 4, \ldots, r_i$, if $\tau(u_i) = true$, or for $k = 1, 3, \ldots, r_{i-1}$, if $\tau(u_i) = false$. Let us call the former cliques as *true cliques*, and the latter ones as *false cliques*. Now, since each Q_j is satisfied by at least one literal, say x, then at least one of the following cases occur: either x is positive, say $x = u_i$, and thus the vertex v corresponding to D_x can be added to the true clique corresponding to C^i_k and C^i_ℓ; or $x = \overline{u_i}$, and thus v can be added to the false clique corresponding to C^i_k and C^i_ℓ. In either case, only two additional cliques need to be added to $\mathcal{Z}$ to cover the remaining vertices associated to T^j (see the cases in Fig. 6 again). Therefore, $|\mathcal{Z}| = p$.

Suppose now that $\mathcal{Z}$ is a clique partition of G' with $|\mathcal{Z}| = p$. Let us construct a truth assignment τ for $\mathcal{Q}$. Since $\mathcal{Z}$ contains no more than p cliques, it is easy to see that the r_i vertices corresponding to C^i are covered by exactly $\frac{r_i}{2}$ cliques, for every i, and the five vertices corresponding to T^j are covered by exactly two cliques, for every j. This means that in every junction T^j one of the disks D_x, D_y, D_z is such that its associated vertex shares a clique with two vertices corresponding to disks C^i_k and C^i_ℓ of some circuit C^i (observe the cases in Figs. 7 and 8). Then, if u_i occurs positively in Q_j, set $\tau(u_i) = true$, otherwise set $\tau(u_i) = false$. In so doing, every clause Q_j contains at least one literal with value true. □

3. Approximation algorithms

The algorithms described in this section assume that the input graph $G = (V, E)$ is a unit disk graph and a realization of G is given. Write $|V| = n, |E| = m$ and $\overline{m} = \binom{n}{2} - m$. We may assume that the realization of G uses $O(n)$ area (in the extreme case G is an edgeless graph, which clearly has an $O(n)$ area realization). Denote by $N(v)$ the set of neighbors of v in G, and by $x(v), y(v)$ the coordinates of the disk associated to vertex v in the realization of G.

3.1. Approximation algorithm for finding clique partitions in unit disk graphs

The approximation algorithm presented in this subsection uses as a subroutine an exact algorithm for finding optimal clique partitions in *k-strip graphs* for $k = \frac{\sqrt{3}}{2}$. A unit disk graph H is a k-strip graph if there exists a real value y_0 such that the centers of the disks in a realization of H are contained in the region $S_k = \{(x, y) \in \Re^2 \mid y_0 \le y < y_0 + k\}$. When $k \le \frac{\sqrt{3}}{2}$, H is also a cocomparability graph [3]. Therefore, it is easy in this case to find a minimum clique partition of H by simply coloring its complement $\overline{H}$; this can be done in time linear in the size of $\overline{H}$ [13,14].

Algorithm 1 Find an approximate clique partition of a unit disk graph G (a realization of G is given)

1: consider a partition of the plane into horizontal strips of width $\frac{\sqrt{3}}{2}$
2: associate to each strip i a subgraph G_i induced by the vertices of G corresponding to the disks whose centers lie in strip i
3: find an exact clique partition $\mathcal{Z}_i$ for each G_i
4: return a clique partition $\mathcal{Z} = \cup \mathcal{Z}_i$

We assume without loss of generality that, in the realization of G, all disk centers have nonnegative y-coordinates, and the bottommost disk centers have

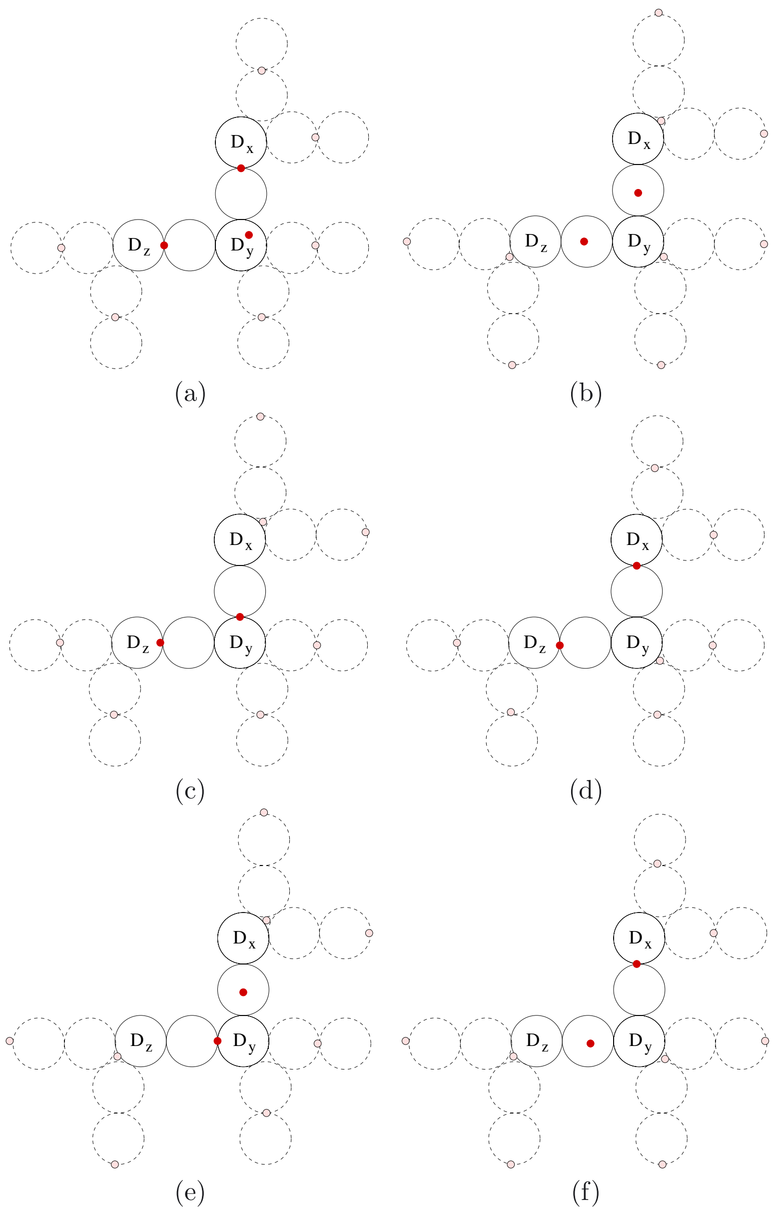
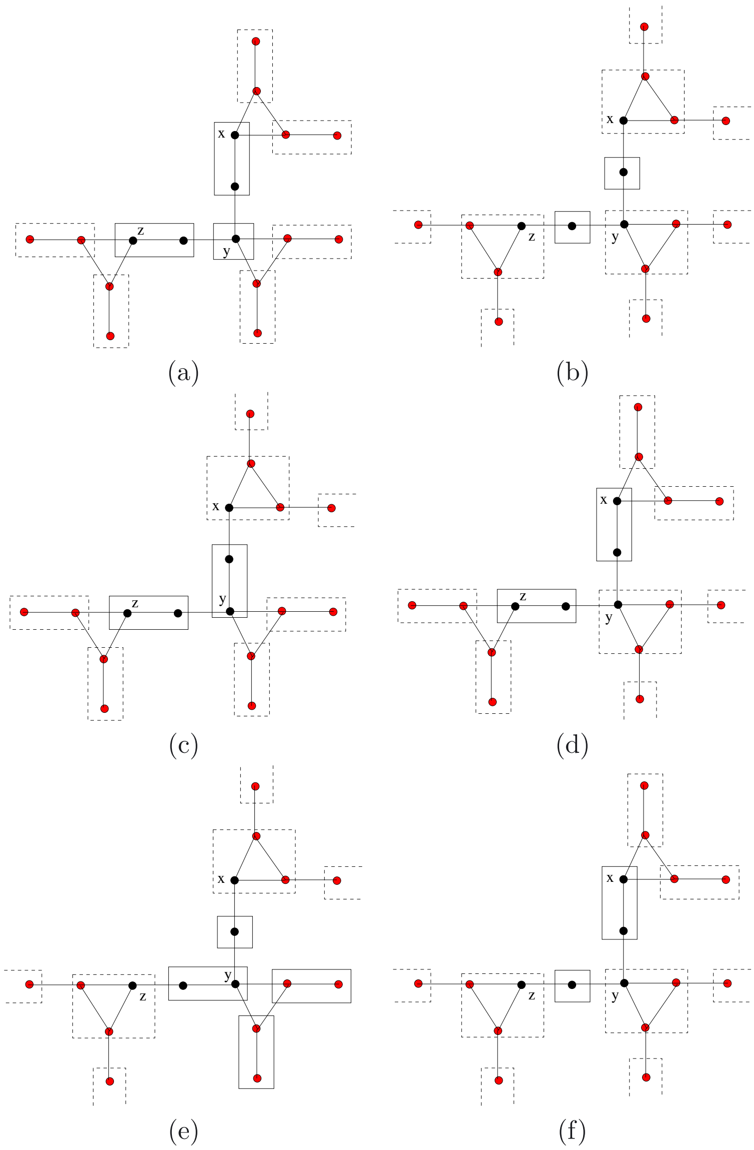


FIGURE 7. Covering a junction by cliques (represented here by points): (a) D_x, D_y, D_z not covered by cliques in circuits; (b) D_x, D_y, D_z already covered; (c) D_x already covered; (d) D_y already covered; (e) D_x and D_z already covered; (f) D_y and D_z already covered. Remaining cases are analogous.



(a) (b) (c) (d) (e) (f)

Figure 8. Covering a clause by cliques (represented here by rectangles): (a) Q_j is not satisfied; (b) Q_j satisfied by literals x, y and z; (c) Q_j satisfied by x; (d) Q_j satisfied by y; (e) Q_j satisfied by x and z; (f) Q_j satisfied by y and z. Remaining cases are analogous. Dotted rectangles enclose cliques used for covering circuits; continuous rectangles enclose cliques used for covering clauses.

y-coordinates equal to zero. In Line 1, the plane is partitioned into ℓ horizontal strips of width $\frac{\sqrt{3}}{2}$ each, where $\ell = \lceil \frac{\overline{y}}{\sqrt{3}/2} \rceil$ and $\overline{y}$ is the maximum value of an y-coordinate considering the disk centers. Every disk whose center $c = (x, y)$ satisfies $(i-1)\frac{\sqrt{3}}{2} \le y < i\frac{\sqrt{3}}{2}$ belongs to strip i, $1 \le i \le \ell$.

For each vertex v, determining which horizontal strip the point $(x(v), y(v))$ lies in takes $O(1)$ time; thus, Line 2 take $O(n)$ time (recall that any two disk centers are $O(n)$ apart). Line 3 takes $O(n + \overline{m})$ time: if G_i contains n_i vertices and m_i edges, then $\sum_{1 \le i \le \ell} n_i = n$ and $\sum_{1 \le i \le \ell} m_i \le m$. Thus, the overall running time of Algorithm 1 is $O(n + \overline{m})$, that is, linear in the size of $\overline{G}$.

Let $\mathcal{Z}^*$ be a minimum clique partition of G, and let $\mathcal{Z}_i^*$ be the restriction of $\mathcal{Z}^*$ to G_i, that is, $\mathcal{Z}_i^* = \{C_i \ne \emptyset \mid C_i = C \cap V(G_i) \text{ and } C \in \mathcal{Z}^*\}$. Since Step 3 covers G_i optimally, $|\mathcal{Z}_i| \le |\mathcal{Z}_i^*|$. Now, let C be a clique in $\mathcal{Z}^*$. By a simple geometric argument, the disk centers associated to the vertices of C are distributed along at most three strips (note that, in the extreme case, the disk centers define a right triangle whose legs have lengths $\frac{1}{2}$ and $\frac{\sqrt{3}}{2}$). This implies that C is the union of at most three disjoint cliques belonging to, say, $\mathcal{Z}_{i-1}^*$, $\mathcal{Z}_i^*$ and $\mathcal{Z}_{i+1}^*$. That is, $|\mathcal{Z}| = \sum_{i=1}^{\ell} |\mathcal{Z}_i| \le \sum_{i=1}^{\ell} |\mathcal{Z}_i^*| \le 3|\mathcal{Z}^*|$. Hence Algorithm 1 is a 3-approximation algorithm.

3.2. Approximation algorithm for finding clique partitions of penny graphs

Assume that G is a penny graph. The algorithm in this subsection is a simple greedy heuristic based on the following straightforward facts:

1. If C is a clique in a penny graph, then $|C| \le 3$.
2. If G is also a 1-strip graph, and v is a vertex with $x(v)$ minimum, then $|N(v)| \le 2$.

The strategy of the approximation algorithm is the same as in Algorithm 1.

Algorithm 2 Find an approximate clique partition of a penny graph G (a realization of G is given)

1: consider a partition of the plane into horizontal strips of width 1
2: associate to each strip i a subgraph G_i induced by the vertices of G corresponding to the disks whose centers lie in strip i
3: find an exact clique partition $\mathcal{Z}_i$ for each G_i
4: return a clique partition $\mathcal{Z} = \cup \mathcal{Z}_i$

Again, disk centers have nonnegative y-coordinates and the bottommost disk centers have y-coordinates equal to zero. If $y(v)$ satisfies $i - 1 \le y(v) < i$, for some $i \ge 0$, then v is a vertex of G_i.

The exact clique partition $\mathcal{Z}_i$ for G_i in Line 3 is obtained as follows. Let $v_1^i, \ldots, v_{k_i}^i$ be an ordering of the vertices of G_i such that $x(v_j^i) \le x(v_{j+1}^i)$, for $1 \le j < k_i$. We use the lemma below.

Lemma 3.1. *Let C_1^i be a maximal clique containing v_1^i. Then C_1^i belongs to some minimum clique partition of G_i.*

Proof. If $N(v_1^i)$ is a clique then the lemma clearly holds. Otherwise, by Facts 1 and 2, $N(v_1^i) = \{v_2^i, v_3^i\}$ and v_2^i, v_3^i are not adjacent. In this case, either $\{v_1^i, v_2^i\}$ or $\{v_1^i, v_3^i\}$ belongs to some minimum clique partition of G_i. □

Lemma 3.1 leads to a greedy method to compute $\mathcal{Z}_i$: repeatedly (i) find a maximal clique containing the leftmost vertex in the ordering, and (ii) remove the vertices of such a clique from the current graph. This procedure is repeated until no more vertices are left.

Line 2 take $O(n)$ time. By considering a partition of the plane into vertical strips of the form $j \le x < j+1$, $j \in \mathbb{Z}$, determining which vertical strip a disk center lies in is immediate. In addition, at most three vertices of G_i can be associated with a same vertical strip j (no four disk centers may simultaneously lie in j because centers located at $x = j + 1$ lie in the next strip). This means that ordering vertices associated with a same vertical strip takes $O(1)$ time. Thus, the ordering $v_1^i, \ldots, v_{k_i}^i$ of the vertices of G_i can be obtained in $O(n_i)$ time, where n_i is the number of vertices of G_i. Finding a maximal clique C_1^i containing v_1^i, according to Lemma 3.1, takes $O(1)$ time, and thus the greedy method to compute $\mathcal{Z}_i$ also takes $O(n_i)$ time. Overall, Line 3 takes $O(n)$ time, and therefore Algorithm 2 takes $O(n)$ time.

It is easy to see that disk centers associated to the vertices of a clique C belonging to a minimum clique partition $\mathcal{Z}^*$ are distributed along at most two horizontal strips of width 1. By applying the same argument as in the previous subsection, Algorithm 2 is a 2-approximation algorithm. This result corrects the previous result in [7].

An interesting question is to devise linear-time algorithms with better approximation ratios than the described ones.

References

[1] B.S. Baker, Approximation algorithms for NP-complete problems on planar graphs. *J. ACM* **41** (1994) 153–180.

[2] P. Berman, M. Karpinski and A.D. Scott, Approximation hardness and satisfiability of bounded occurrence instances of SAT, in *Electronic Colloquium on Computational Complexity – ECCC* (2003).

[3] H. Breu, *Algorithmic Aspects of Constrained Unit Disk Graphs*. Ph.D. thesis, University of British Columbia (1996).

[4] H. Breu and D.G. Kirkpatrick, Unit disk graph recognition is NP-hard. *Computational Geometry* **9** (1998) 3–24.

[5] A. Borodin, I. Ivan, Y. Ye and B. Zimny, On sum coloring and sum multi-coloring for restricted families of graphs. Manuscript available at http://www.cs.toronto.edu/∼bor/2420f10/stacs.pdf consulted 30 July 2011.

[6] B.N. Clark, C.J. Colbourn and D.S. Johnson, Unit disk graphs. *Discrete Math.* **86** (1990) 165–177.

[7] M.R. Cerioli, L. Faria, T.O. Ferreira and F. Protti, On minimum clique partition and maximum independent set in unit disk graphs and penny graphs: complexity and approximation. *LACGA'2004 – Latin-American Conference on Combinatorics, Graphs and Applications.* Santiago, Chile (2004). *Electron. Notes Discrete Math.* **18** (2004) 73–79.

[8] T. Erlebach, K. Jansen and E. Seidel, Polynomial-time approximation schemes for geometric graphs, in *Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms* (2000) 671–679.

[9] P. Erdös, On some problems of elementary and combinatorial geometry. *Ann. Math. Pura Appl. Ser.* **103** (1975) 99–108.

[10] D. Lichtenstein, Planar formulae and their uses. *SIAM J. Comput.* **43** (1982) 329–393.

[11] M.R. Garey and D.S. Johnson, The rectilinear Steiner tree problem is NP-complete. *SIAM J. Appl. Math.* **32** (1977) 826–834.

[12] M.R. Garey and D.S. Johnson, *Computers and Intractability: a Guide to the Theory of NP-completeness*. Freeman, New York (1979).

[13] M.C. Golumbic, The complexity of comparability graph recognition and coloring. *Computing* **18** (1977) 199–208.

[14] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York (1980).

[15] H. Harborth, Lösung zu problem 664a. *Elem. Math.* **29** (1974) 14–15.

[16] H.B. Hunt III, M.V. Marathe, V. Radhakrishnan, S.S. Ravi, D.J. Rosenkrantz and R.E. Stearns, NC-Approximation schemes for NP- and PSPACE-Hard problems for geometric graphs. *J. Algor.* **26** (1998) 238–274.

[17] K. Jansen and H. Müller, The minimum broadcast time problem for several processor networks. *Theor. Comput. Sci.* **147** (1995) 69–85.

[18] M.V. Marathe, H. Breu, H.B. Hunt III, S.S. Ravi and D.J. Rosenkrantz, Simple heuristics for unit disk Graphs. *Networks* **25** (1995) 59–68.

[19] T. Matsui, Approximation algorithms for maximum independent set problems and fractional coloring problems on unit disk graphs. In *Discrete and Computational Geometry. Lect. Notes Comput. Sci.* **1763** (2000) 194–200.

[20] I.A. Pirwani and M.R. Salavatipour, A weakly robust PTAS for minimum clique partition in unit disk graphs (Extended Abstract), *Proceedings of SWAT 2010. Lect. Notes Comput. Sci.* **6139** (2010) 188–199.

[21] S.V. Pemmaraju and I.A. Pirwani, Good quality virtual realization of unit ball graphs, *Proceedings of the 15th Annual European Symposium on Algorithms. Lect. Notes Comput. Sci.* **4698** (2007) 311–322.

[22] O. Reutter, Problem 664a. *Elem. Math.* **27** (1972) 19.

[23] J.P. Spinrad, *Efficient Graph Representations*, *Fields Institute Monographs* **19**. American Mathematical Society (2003).

[24] L.G. Valiant, Universality considerations in VLSI circuits. *IEEE Trans. Comput.* **30** (1981) 135–140.

Communicated by C. Choffrut.
Received June 23, 2009. Accepted March 23, 2011.