単位円グラフ上の最大独立集合問題の近似解法

松井 知己
東京大学大学院 工学系研究科 計数工学専攻 113-8656 文京区本郷 7-3-1
tomomi@misojiro.t.u-tokyo.ac.jp

概要： 単位円グラフは，２次元平面上の単位円の交差グラフである．本稿では，単位円グラフ上の最大独立集合問題の解法を提案する．頂点数 n の単位円グラフが，幅 k の横長の板の上に定義されている場合，最大独立集合を見つける $O(n^{4\lceil 2k/\sqrt{3}\rceil})$ 時間の解法を提案する．また一般の単位円グラフに付いては，最大独立集合問題の $(1-1/r)$-近似解法で，時間計算量が $O(rn^{4\lceil 2(r-1)/\sqrt{3}\rceil})$ となるものを提案する．

Approximation algorithms for maximum independent set problems on unit disk graphs

Tomomi Matsui
Department of Mathematical Engineering and Information Physics,
Graduate School of Engineering, University of Tokyo, Bunkyo-ku, Tokyo 113,
Japan.
tomomi@misojiro.t.u-tokyo.ac.jp,

Abstract: Unit disk graphs are the intersection graphs of equal sized circles in the plane. In this paper, we consider the maximum independent set problem on the unit disk graph. When the given unit disk graph is defined on a slab whose width is k, we propose an algorithm for finding a maximum independent set in $O(n^{4\lceil 2k/\sqrt{3}\rceil})$ time where n denotes the nuber of vertices. We also propose a $(1-1/r)$-approximation algorithm for the maximum independent set problems on a (general) unit disk graph whose time complexity is bounded by $O(rn^{4\lceil 2(r-1)/\sqrt{3}\rceil})$.

1 Preliminaries

Unit disk graphs are the intersection graphs of equal sized circles in the plane. Given a point-set $P \subseteq \mathrm{R}^2$ the unit disk graph defined by P, denoted by $G(P)$, is an undirected graph (P, E) with vertex set P and edge set E satisfying that $E = \{\{\boldsymbol{p}_i, \boldsymbol{p}_j\} \mid \boldsymbol{p}_i, \boldsymbol{p}_j \in P, \ ||\boldsymbol{p}_i - \boldsymbol{p}_j|| \leq 1\}$. The unit disk graphs provide a graph-theoretic model for broadcast network and for some problems in computational geometry.

In this paper, we consider the maximum independent set problem on the unit disk graph. A vertex subset P' of a graph is called an independent set if each pair in P' is non-adjacent. The maximum independet set problem finds an independent set in a given graph whose size is maximized. It is well-known that for general graphs, the maximum independent set problem is hard to approximate. Unless $\mathcal{P}$=$\mathcal{NP}$, there exists a constant $\varepsilon > 0$ such that no polynomial time algorithm for the problem can provide a performance guarantee of $\mathrm{O}(n^{\varepsilon})$ where n denote the number of vertices [1]. In 1990, Clark, Colbourn and Johnson [2] proved that the maximum independent set problem defined on unit disk graph is $\mathcal{NP}$-hard. In 1995, Marathe et al. [6] developed a (1/3)-approximation algorithm based on a graph coloring heuristic proposed by Hochbaum in [3]. In this paper, we propose a polynomial time algorithm for the independent set problem on unit disk graphs defined on a fixed width slab. When the width of the slab is k, the time complexity of our algorithm is $\mathrm{O}(|P|^{4\lceil 2k/\sqrt{3}\rceil})$. Our approach also implies an approximation algorithm for the independent set problem on a unit disk graph (defined on a variable width slab), which finds a $(1-1/r)$-approximation solution in $\mathrm{O}(r|P|^{4\lceil 2(r-1)/\sqrt{3}\rceil})$ time. It is easy to extend our algorithm for solving weighted independent set problem on unit disk graph without increasing its time complexity.

2 Maximum independnt set problems

In this section, we consider unit disk graphs defined on a slab whose width is less than k. More precisely, we assume that the point-set P is contained in the region $S_k = \{(x, y) \in \mathrm{R}^2 \mid 0 \leq y < k\}$.

2.1 Well-solvable case

First, we consider a well-solvable case. If $P \subseteq S_k$ and $k < \sqrt{3}/2$, we can solve the independent set problem on the unit disk graph $G(P)$ in polynomial time.

The following lemma shows an idea to solve the problem.

Lemma 1 *If $P \subseteq S_k$ and $k < \sqrt{3}/2$, then the unit disk graph $G(P)$ is a co-comparability graph (the complement of a comparability graph).*

Proof: Let $\overline{G}(P)$ be the complement of the unit disk graph $G(P)$. We direct each edge in $\overline{G}(P)$ as follows. Let e be an edge in $\overline{G}(P)$ connecting two vertices (points)

$p_1, p_2 \in P$. Without loss of generality, we can assume that the x-coordinate of p_1 is less than that of p_2, since the width of the slab is less than 1. We direct the edge e from p_1 to p_2. Now we show that the obtained directed graph, denoted by $\overline{A}(P)$, satisfies the transitivity. Clearly, $\overline{A}(P)$ is acyclic. Assume that $\overline{A}(P)$ contains a pair of directed edges $e = (p_1, p_2)$ and $f = (p_2, p_3)$. We denote the position of p_i by (x_i, y_i) for $i = 1, 2, 3$. Since $||p_1 - p_2|| > 1$ and the width of the slab is less than $\sqrt{3}/2$, it is eay to show that $x_1 + 1/2 < x_2$. In the same way, we can show that $x_2 + 1/2 < x_3$. It implies that $x_1 + 1 < x_3$ and so $||p_1 - p_3|| > 1$. It implies that the complement $\overline{G}(P)$ contains the edge $\{p_1, p_3\}$ and the above edge orientation procedure directs the edge from p_1 to p_3. Thus the directed graph $\overline{A}(P)$ satisfies the transitivity and so the complement $\overline{G}(P)$ is a transitivity graph. //

The co-comparability graph is a class is perfect graphs and so we can solve the maximum independent set problem, the coloring problem, and the maximum clique problem in polynomial time [4]. However, we can solve the maximum independent set problems easily in this case. The proof of the above lemma shows that each directed path in the graph $\overline{A}(P)$ corresponds to an independent set of the unit disk graph $G(P)$. It is also easy to show that each independent set in $G(P)$ corresponds to the vertices in a directed path in $\overline{A}(G)$. Thus, the maximum independent set problem is reduced to the problem for finding longest directed path in $\overline{A}(P)$. Since $\overline{A}(P)$ is acyclic, we can solve the longest path problem in linear time with respect to the number of directed edges. Thus we can find a maximum independent set of $G(P)$ in $O(|P|^2)$ time.

2.2 Fixed width problem

Here we assume that the width k of the slab is a fixed constant.

For any point subset $P' \subseteq P$, we denote the value $\min\{\lfloor x \rfloor \mid (x, y) \in P'\}$ by $\min P'$. A subset of points $B \subseteq P$ is called an *independent block* when B is an independent set of the unit disk graph $G(P)$ and each point (x, y) in P' satisfies $\lfloor x \rfloor = \min B$. Let $\mathcal{B}(P)$ be the family of all the independent blocks of P.

Now we introduce an auxiliary graph which is helpful for finding a maximum independent set of $G(P)$. The auxiliary graph, denoted by $A(P)$, is a directed graph with node set $\{s, t\} \cup \mathcal{B}(P)$ and arc (directed edge) set

$$\{(s, B) \mid \forall B \in \mathcal{B}(P)\} \cup \{(B, t) \mid \forall B \in \mathcal{B}(P)\}$$
$$\cup \{(B, B') \in \mathcal{B}(P) \times \mathcal{B}(P) \mid (\min B) < (\min B') \text{ and } B \cup B' \text{ is an independent set}\}.$$

Then it is clear that for any directed path in the auxiliary graph from s to t, the union of independent blocks corresponding to internal nodes is an independent set of $G(P)$. Conversely, for any independent set in $G(P)$ there exists a corresponding directed s-t path in $A(P)$. For each non-terminal node (independent block) of the auxiliary graph, we associate the weight which is equal to the size of the corresponding independent block. Then the sum of the node weights in a directed path is equal to the size of corresponding independent set in $G(P)$. Thus, the maximum independent set problem on$G(P)$ is reduced to the problem for finding the longest directed path in the auxiliary graph.

We can generate all the indepent blocks by applying an enumeration algorithm for maximal independent sets in [7] which requires $O(|P|^3|\mathcal{B}(P)|)$ time. Since the weighted auxiliary graph is acyclic and directed, the ordinary dynamic programming method finds the longest path in linear time with respect to the number of arcs (see the algorithm for the shortest path problem on acyclic graph in [5], for example). From the above, the total time complexity of our algorithm is bounded by $O(|P|^3|\mathcal{B}(P)| + |\mathcal{B}(P)|^2)$. When we denote the size of maximum independent block by α, $|\mathcal{B}(P)| = O(|P|^\alpha)$. If we consder the non-trivial problem instances satisfying that $\alpha \geq 2$, the total time complexity of our algorithm is bounded by $O(|P|^{2\alpha})$

The following lemma shows a simple upper bound of the value α.

Lemma 2 *We define the value α_k by*

$$\alpha_k = \max\{|P'| \mid P' \subseteq [0,1) \times [0,k), \ \forall \boldsymbol{p}_i, \forall \boldsymbol{p}_j \in P', \boldsymbol{p}_i \neq \boldsymbol{p}_j \Rightarrow ||\boldsymbol{p}_i - \boldsymbol{p}_j|| > 1\}$$

Then $\alpha_k \leq 2\left\lceil 2k/\sqrt{3} \right\rceil$.

Proof: Let T be the rectangle $\{(x,y) \mid \sqrt{3}/2 \geq y \geq 0, \ 1 \geq x \geq 0\}$. Then the length of a diagonal line of T is equal to 1. Thus the distance of any pair of points in T is less than or equal to 1. It is clear that the region $[0,1) \times [0,k)$ can be covered by $2\left\lceil 2k/\sqrt{3} \right\rceil$ copies of the rectangle T. Each copy of the rectangle contains at most one points of each independent set. //

The above upper bound implies that the time complexity of our algorithm is bounded by $O(|P|^{4\lceil 2k/\sqrt{3} \rceil})$ when we apply our algorithm to the problem defined on the slab whose width is equal to k.

Lastly, we consider the memory space. The naive implementation of the above algorithm requires the memory space to maintain the auxiliary graph. However, the layerd structure of the auxiliary graph implies that we only need to maintain the nodes of the auxiliary graph.

For any integer k', we denote the set of independent blocks $\{B \in \mathcal{B}(P) \mid k' = \min B\}$ by $\mathcal{B}_k(P)$. Without loss of generality, we can assume that $\mathcal{B}(P)$ can be partitioned into the families of independent blocks $\mathcal{B}_0(P), \mathcal{B}_1(P), \ldots, \mathcal{B}_m(P)$. Also we can asuume that $\mathcal{B}_{k'}(P) \neq \emptyset$ for all $k' \in \{0, \ldots, m\}$, since if there exists a family $\mathcal{B}_{k'}(P) = \emptyset$ we can decompose the original problem to two small subproblems defined by the sets of points $\mathcal{B}_0(P) \cup \cdots \cup \mathcal{B}_{k'-1}(P)$ and $\mathcal{B}_{k'+1}(P) \cup \cdots \cup \mathcal{B}_m(P)$. Under the above assumption, every maximal independent set of $G(P)$ contains at least one independent block $B \in \mathcal{B}_{k'-1}(P) \cup \mathcal{B}_{k'}(P)$ for all $k' \in \{1, \ldots, m\}$. Thus we can delete arcs in the auxiliry graph without sacrificing the correctness of our algorithm, i.e., we only need the following arcs

$$\begin{aligned}&\{(s, B) \mid \forall B \in \mathcal{B}(P)\} \cup \{(B, t) \mid \forall B \in \mathcal{B}(P)\}\\ &\cup\{(B, B') \mid (\min B) + 1 \leq (\min B') \leq (\min B) + 2 \text{ and } (B, B') \text{ is an arc in} A(P)\}.\end{aligned}$$

When we apply the ordinary labeling procedure for solving the longest path problem on $A(P)$ (see [5] for example), we only need to maintain consecutive three families of independent blocks $\mathcal{B}_{k'-1}(P) \cup \mathcal{B}_{k'}(P) \cup \mathcal{B}_{k'+1}(P)$ for labeling independent blocks in $\mathcal{B}_{k'+1}(P)$. When we label an independent block B in $\mathcal{B}_{k'+1}(P)$, we generate arcs in $A(P)$ entering to B one by one. and so we do not need to maintain set of arcs connecting independent blocks in $\mathcal{B}_{k'-1}(P) \cup \mathcal{B}_{k'}(P) \cup \mathcal{B}_{k'+1}(P)$

2.3 Approximation algorithm

In the following, we propose $(1 - 1/r)$-approximation algorithm for any positive integer r, which finds an independent set whose size is greater than or equal to $(1 - 1/r)z^*$ where z^* is the size of a maximum independent set.

For any $s \in \{0, 1, \ldots, r-1\}$, the region $\{(x, y) \in \mathrm{R}^2 \mid s \leq (y \bmod r) < s + 1\}$ is denoted by T_s. .We construct point-subsets $P_0, P_1, \ldots, P_{r-1}$ defined by $P_s = P \setminus T_s$. Next we solve the maximum independent set problems defined on the graphs $G(P_0), G(P_1), \ldots, G(P_{r-1})$ and output one of the best solutions. The size of the output independent set is greater than or equal to $(1 - 1/r)z^*$. It is because, a maximum independent set P^* satisfies that for any $s \in \{0, 1, \ldots, r-1\}$, $P^* \setminus T_s \subseteq P \setminus T_s$ and $\max\{|P^* \setminus T_s| \mid s \in \{0, 1, \ldots, r-1\}\} \geq (1 - 1/r)|P^*|$.

By using the simple upper bound in the previous lemma, the time complexity of our algorithm is bounded by $\mathrm{O}(r|P|^{2\alpha_r - 1}) = \mathrm{O}(r|P|^{4\lceil 2(r-1)/\sqrt{3}\rceil})$.

It is easy to extend our algorithm for the weighted version; a problem defined by the point-set P and point-weights. By substituting the node weights of the auxiliary graph by the sum of weights of points in the corresponding independent block, the algorithm in the previous subsection finds a maximum weight independent set in the same time complexity. The algorithm described in this subsection finds an approximate solution in the same time complexity.

References

[1] S.Arora and C.Lund and R.Motwani and M.Sudan and M.Szegedy, Proof verification and intractability of approximation problems, *Proc. 33rd IEEE Symposium on Foundations of Computer Sciece*, pp.13-22, 1992.

[2] B.N.Clark and C.J.Colbourn and D.S.Johnson, Unit disk graphs, *Discrete Mathematics*, 86 (1990), pp.165-177.

[3] D.S.Hochbaum, Efficient bounds for the stable set, vertex cover and set packing problems, *Discrete Applied Mathematics*, 6 (1983), pp.243-254.

[4] M. Grötschel and L. Lovász and A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica*, 1 (1981), pp.169-197.

[5] E.L. Lawler, *Combinatorial Optimication: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.

[6] M.V.Marathe and H.Breu and H.B. Hunt III and S.S. Ravi and D.S. Rosenkrantz, Simple heuristics for unit disk graphs, *Networks*, 25 (1995), pp.59-68.

[7] S. Tsukiyama and M. Ide and H. Ariyoshi and I. Shirakawa, A new algorithm for generating all the maximal independent sets, *SIAM J. on Computing*, 6 (1977), pp.505-517.