A Robust PTAS for Maximum Weight Independent Sets in Unit Disk Graphs

Tim Nieberg*, Johann Hurink, and Walter Kern

University of Twente
Faculty of Electrical Engineering, Mathematics & Computer Science
Postbus 217, NL-7500 AE Enschede
{T.Nieberg,J.L.Hurink,W.Kern}@utwente.nl

Abstract. A unit disk graph is the intersection graph of unit disks in the euclidean plane. We present a polynomial-time approximation scheme for the maximum weight independent set problem in unit disk graphs. In contrast to previously known approximation schemes, our approach does not require a geometric representation (specifying the coordinates of the disk centers).
The approximation algorithm presented is robust in the sense that it accepts any graph as input and either returns a $(1+\varepsilon)$-approximate independent set or a certificate showing that the input graph is no unit disk graph. The algorithm can easily be extended to other families of intersection graphs of geometric objects.

1 Introduction

A unit disk graph (*UDG*) is the intersection graph of unit disks in the plane. In other words, $G = (V, E)$ is a UDG if there exists a map $f : V \to \mathbb{R}^2$ (a *geometric representation*) satisfying

$$(u, v) \in E \iff \|f(u) - f(v)\| \leq 2, \tag{1}$$

where $\|.\|$ denotes the euclidean norm.

A subset of vertices in G is called independent if the vertices in this subset are pairwise not connected by an edge. The maximum independent set problem now consists of finding a such an independent subset of the vertices of maximum cardinality. For the maximum weight independent set problem, each vertex $v \in V$ is also assigned a weight $w_v > 0$, and the goal is to find an independent set of maximal total weight, i.e. of maximum sum of all weights of the vertices in the independent set.

In this document, we give a *polynomial-time approximation scheme* (PTAS) for the maximum (weight) independent set problem in unit disk graphs for the case that a geometric representation is not given, i.e. we seek for an algorithm which, given as input a UDG $G = (V, E)$ and a parameter $\varepsilon > 0$, computes an

* This work is partially supported by the European research project EYES (IST-2001-34734).

J. Hromkovič, M. Nagl, and B. Westfechtel (Eds.): WG 2004, LNCS 3353, pp. 214–221, 2004.

independent set $I \subset V$ of size (weight) at least $(1+\varepsilon)^{-1}$ times the maximum size (weight) of an independent set in G. The running time of the algorithm is allowed to depend on ε, but should be polynomial in $n = |V|$ for fixed $\varepsilon > 0$.

Most of the work concerning approximation schemes in unit disk graphs has been done assuming a given geometric representation. The representation makes it possible to perform separation of the graph alongside a grid. This so called *shifting strategy* is presented in [1] and [7]. Combined with a dynamic programming approach, the shifting strategy is used by Erlebach et. al. [5] to give a PTAS for the maximum weight independent set in disk graphs of arbitrary diameter. Using quadtrees as separation, the runtime has been improved by Chan [3] to $n^{O(1/\varepsilon^{d-1})}$, where d gives the dimension of the euclidean space. The shifting strategy can also be applied to related problems like minimum vertex cover, and minimum dominating set [8]. Also, the minimum connected dominating set problem can be approximated by a PTAS with the help of separation using a grid structure [4].

The case where no geometric representation f for the UDG $G = (V, E)$ is available is significantly different: Computing a corresponding representation function f for a given UDG $G = (V, E)$ is NP-hard. Indeed, any polynomial time algorithm computing geometric representation functions for unit disk graphs could be used in a straightforward way to solve the UDG recognition problem (determine whether a given graph is a UDG), which is known to be NP-hard [2].

Without given geometric representation, a PTAS for the maximum (weight) independent set problem was not known previously. However, constant factor approximation algorithms have been given in the literature. A simple greedy strategy gives a 5-approximation for the maximum weight independent set, and a more sophisticated choice of a node to be greedily added to the partial set of independent nodes gives an approximation within a factor of 3 for the unweighted case [10]. Both algorithms work without given representation, however, the running time can be improved a lot when the representation is given.

Unit disk graphs form a subclass of the more general class of all undirected graphs. This raises the question of robustness (see [12]) for algorithms designed for the restricted domain of UDGs. Generally speaking, a robust algorithm $\mathcal{A}$ on a restricted class $\mathcal{U} \subset \mathcal{G}$ solves a problem for all instances in $\mathcal{U}$, but also accepts any instance in $\mathcal{G}$. For instances in $\mathcal{G} \setminus \mathcal{U}$, the algorithm $\mathcal{A}$ either solves this problem or provides a certificate showing that the input is not in $\mathcal{U}$. For the PTAS presented in this paper, the algorithm accepts any graph as input, e.g. given by an adjacency list or matrix, and either returns a $(1+\varepsilon)$-approximate (weighted) independent set or a certificate to show that the graph does not belong to the class of unit disk graphs. In case the input graph is a unit disk graph, the algorithm always returns an independent set of desired quality.

The problem of finding a maximum (weight) independent set arises for example in the context of clustering in wireless ad-hoc networks [11]. The battery-operated nodes, each equipped with a radio transceiver of fixed transmission range, form a UDG representing the communication network. Nodes of a maximum independent set can be used as controlling instances, or clusterheads, of

the other nodes within range of their radio. A maximum independent set also forms a dominating set. In the ad-hoc scenario, it is hard or costly to determine the exact position of each node and therefore for the resulting network only a representation by the nodes and communication links between them is known. The weights of those nodes may correspond to the residual energy in order to make nodes with more battery power clusterheads to prolong the operational lifetime of the network.

The remainder of this paper is organized as follows. The next section introduces the algorithm that gives the PTAS for the unweighted case of finding an independent set of maximum cardinality in UDGs without using a geometric representation. Section 3 then gives the modifications of the algorithm to efficiently approximate the maximum weight independent set. The algorithms are presented with the assumption (or "promise") that the input is a UDG. In Section 4, we show how to obtain a robust algorithm from the PTAS. In Section 5, we identify some other classes of geometric intersection graphs for which our approach also is efficient. The paper ends with a short conclusion and an outlook on future work.

2 The Approximation Algorithm

In this section, we introduce the approximation algorithm that forms the core of the robust PTAS for the maximum independent set problem on UDGs. The same algorithm is then adapted to the weighted version of the problem in Section 3.

The algorithm does not rely on a geometric representation, it thus accepts any graph as an input-instance. However, the statements concerning the running time depend on the assumption that the graph has a geometric representation.

Let $\varepsilon > 0$ and let $\rho := 1 + \varepsilon$ denote the desired approximation guarantee. Thus, given a unit disk graph $G = (V, E)$, we seek to construct an independent set $I \subset V$ of cardinality at least ρ^{-1} times $\alpha(G)$, the maximum size of an independent set in G.

The basic idea is simple. We start with an arbitrary node $v \in V$ and consider for $r = 0, 1, 2, \ldots$, the r^{th} neighborhood

$$N^r = N^r(v) := \{w \in V | w \text{ has distance at most } r \text{ from } v\}.$$

Starting with N^0, we compute a maximum independent set $I_r \subset N^r$ for each $r = 0, 1, 2, \ldots$ as long as

$$|I_{r+1}| > \rho |I_r| \tag{2}$$

holds.

Let $\bar{r}$ denote the smallest $r \geq 0$ for which (2) is violated. Such an $\bar{r} \geq 0$ indeed exists and it is bounded by a constant (depending on ρ):

Lemma 1. *There exists a constant $c = c(\rho)$ such that $\bar{r} \leq c$.*

Proof. From (1), we conclude that any $w \in N^r$ satisfies

$$\|f(v) - f(w)\| \leq 2r.$$

So, the unit disks corresponding to nodes in I_r are pairwise disjoint and are all contained in a disk of radius $R = 2r + 1$ around $f(v)$. This implies

$$|I_r| \leq \pi R^2/\pi = O(r^2). \tag{3}$$

On the other hand, by definition of $\bar{r}$, we have for $r < \bar{r}$

$$|I_r| > \rho|I_{r-1}| > \ldots > \rho^r|I_0| = \rho^r. \tag{4}$$

Comparing (3) and (4), the claim follows. □

To achieve an independent set for the graph G, the above algorithm is iteratively applied to the graph $G' := G \setminus N^{\bar{r}+1}$, and the resulting independent set for G' is combined with $I_{\bar{r}}$ for an independent set in G. Note that, due to (3), we may compute I_r by complete enumeration in time $O(n^{C^2})$, where $C = O(r) = O(1/\varepsilon^2 \log 1/\varepsilon)$ for $r \leq \bar{r}$ (see Appendix). The algorithm evolving from the above description thus runs in polynomial time, as all other computations are dominated by this complexity. The correctness and approximation guarantee of the algorithm follows from the following theorem.

Theorem 1. *Suppose inductively that we can compute a ρ-approximate independent set $I' \subset V \setminus N^{\bar{r}+1}$ for G'. Then $I := I_{\bar{r}} \cup I'$ is a ρ-approximate independent set for G.*

Proof. Since each $v \in I' \subset V \setminus N^{\bar{r}+1}$ has no neighbor in $N^{\bar{r}}$, and thus not in $I_{\bar{r}} \subset N^{\bar{r}}$, I is an independent set.
Furthermore, by definition of $\bar{r}$, we have

$$|I_{\bar{r}+1}| \leq \rho|I_{\bar{r}}|.$$

In other words, the subgraph $G[N^{\bar{r}+1}]$ induced by $N^{\bar{r}+1}$ has a maximum independent set size bounded by

$$\alpha(G[N^{\bar{r}+1}]) \leq \rho|I_{\bar{r}}|.$$

Further, by assumption, I' is ρ-approximately optimal for $G' = G[V \setminus N^{\bar{r}+1}]$. Thus,

$$\alpha(G[V \setminus N^{\bar{r}+1}]) \leq \rho|I'|.$$

Adding the two inequalities, we obtain

$$\alpha(G) \leq \alpha(G[N^{\bar{r}+1}]) + \alpha(V \setminus G[N^{\bar{r}+1}]) \leq \rho|I|,$$

as claimed. □

3 The Algorithm for the Weighted Problem

The approximation algorithm presented in the previous section for the maximum independent set problem on UDGs can easily be adapted for the case that each

node $v \in V$ is also given a nonnegative weight w_v. In that case, we are seeking an independent set of maximum total weight in the unit disk graph G. In the following, we present the modified algorithm that returns an independent set of total weight at least $(1+\varepsilon)^{-1}$ the maximum total weight of an independent set in the UDG given as input.

For a subset $I \subset V$ of vertices, let $W(I)$ denote the total weight of I, i.e.$W(I) = \sum_{i \in I} w_i$. Furthermore, let I^{OPT} be an optimal solution to the maximum weight independent set problem for the graph $G = (V, E)$.

The approximation algorithm again follows the idea of the algorithm in the previous section. This time, however, we start with a vertex of maximal weight $w_{\max} = \max\{w_i | i \in V\}$, and then compute the independent set $I_r \subset N^r$ of maximum weight as long as $W(I_{r+1}) > \rho W(I_r)$ holds. Let $\bar{r}$ denote the smallest $r \geq 0$ for which this criterion is violated.

Lemma 2. *There exists a constant $c = c(\rho)$ such that $\bar{r} \leq c$.*

Proof. Suppose $r < \bar{r}$. Adapting the proof of Lemma 1, we get

$$W(I_r) = \sum_{i \in I_r} w_i \leq \sum_{i \in I_r} w_{\max} = |I_r| w_{\max}, \tag{5}$$

and

$$W(I_r) > \rho W(I_{r-1}) > \ldots > \rho^r W(I_0) = \rho^r w_{\max} \tag{6}$$

respectively. Since $|I_r| = O(r^2)$, comparing (5) and (6) again yields the claim. □

The running time of this algorithm remains polynomial in the weighted case. Also, the approximation ratio can be guaranteed as follows.

Theorem 2. *The adapted algorithm yields an independent set of weight at least $\rho^{-1} = (1+\varepsilon)^{-1}$ the weight of a maximum weight independent set.*

Proof. Let $V' := V \backslash N^{\bar{r}+1}$, and inductively assume $I' \subset V'$ to be a ρ-approximate independent weighted set in $G[V']$. Clearly, $I_{\bar{r}} \cup I'$ is an independent set in G. For the weighted independent set in the neighborhood $N^{\bar{r}+1}$, we have

$$W(I^{\mathrm{OPT}} \cap N^{\bar{r}+1}) \leq W(I_{\bar{r}+1}) \leq \rho W(I_{\bar{r}}).$$

For the weight of the set returned by the algorithm, $W(I_{\bar{r}} \cup I')$, it is

$$\begin{aligned} W(I^{\mathrm{OPT}}) &= W((I^{\mathrm{OPT}} \cap N^{\bar{r}+1}) \cup (I^{\mathrm{OPT}} \cap V')) \\ &= W(I^{\mathrm{OPT}} \cap N^{\bar{r}+1}) + W(I^{\mathrm{OPT}} \cap V') \\ &\leq \rho W(I_{\bar{r}}) + \rho W(I') \\ &= \rho W(I_{\bar{r}} \cup I'). \end{aligned}$$ □

4 Robustness

In this section, we show that the approximation algorithms of the previous two sections actually lead to a robust algorithm.

Definition 1. *Let $\mathcal{A}$ be an algorithm defined on $\mathcal{G}$, f be a function on $\mathcal{G}$, and $\mathcal{U} \subset \mathcal{G}$. Then $\mathcal{A}$ computes f robustly (on $\mathcal{U}$), if*

1. *for all instances $i \in \mathcal{U}$, the algorithm $\mathcal{A}$ returns $f(i)$, and*
2. *for all instances $i \in \mathcal{G} \backslash \mathcal{U}$, the algorithm $\mathcal{A}$ returns either $f(i)$, or a certificate showing that $i \notin \mathcal{U}$.*

Of course, the notion of a robust algorithm is especially interesting when $\mathcal{A}$ has polynomial running time with respect to the size of the input instance, and the decision whether an instance belongs to the subclass $\mathcal{U} \subset \mathcal{G}$ is not as easy to decide. In our situation, $\mathcal{G}$ is the set of all undirected graphs, f gives a $(1+\varepsilon)$-approximation of the weight or cardinality of an independent set of maximum weight or size respectively, and $\mathcal{U}$ is the subclass of unit disk graphs.

In the previous sections, we have shown that the introduced approximation algorithms yield a PTAS for the case that the input instances represent a unit disk graph. We thus continue our discussion only for the case that the input instance is a graph for which there exists no geometric representation satisfying the characterization of a UDG.

Observe that in Theorems 1 and 2, we did not use any properties of a UDG. The algorithms thus always return a $(1+\varepsilon)$-approximate independent set. However, the polynomial running time of the algorithms is a direct result from the fact that any independent set in the r^{th} neighborhood is polynomially bounded in r, i.e. $|I_r| \leq (2r+1)^2$. For a general graph, the running time may thus not be polynomial. So, during the execution of the algorithm, if an independent set of size $|I_r| > (2r+1)^2$ can be found, this set is returned as certificate of non-membership in the class of unit disk graphs. This procedure, i.e. trying to find an independent set of size $(2r+1)^2+1$, is also a task requiring polynomial running time.

5 Extensions

It is straightforward to verify that our arguments apply equally well to intersection graphs of some other geometrical objects related to unit disks. For example, the unit disks may be replaced by disks with fixed lower and upper bound on the radius (*bounded disk graphs*). Similarly, an extension to (fixed) dimension $d \geq 2$ is possible. Indeed, all that is needed in the proof is a polynomial bound on the maximum geometric diameter divided by the minimum volume of the objects under consideration. Quasi (Unit) Disk Graphs, as a more realistic model of a wireless communication graph [9] satisfy this characterization.

The algorithm can also be applied to λ-precision disk graphs though they have no bound on the radius of the disks and thus no bound on the minimum volume of the disks. A λ-precision disk graph is an intersection disk graph where two vertices are at least λ apart in a geometric representation that has been scaled so that the disks have a maximum radius of 1 [8]. In this case, the size of the r^{th} neighborhood, $|N^r|$, is already polynomially bounded in r. Note that this is a different condition as the one given above: for example, in a UDG, two vertices may be arbitrarily close as the graph can contain arbitrarily large cliques.

The independent set created by the PTAS may not be maximal, i.e. there may exist vertices that can be added to the solution set returned by the PTAS without violating the independent set property. However, a simple greedy strategy on the nodes in $N^{r_1+1} \setminus N^{r_1}$ that are not connected by an edge to a node from the independent set can resolve this problem. The obtained independent set then also forms a dominating set in the graph.

6 Conclusion

In this paper, we present a new PTAS for the maximum independent set problem in UDGs that does not depend on a geometric representation of the vertices. The algorithm is extended to solve also the weighted version of the problem. Both algorithms accept any graph as input and either return a $(1 + \varepsilon)$-approximate independent set or a certificate showing that the input presented to the algorithm is not a UDG. This certificate is given by an independent set which is too large to be contained in a bounded area given by the neighborhood N^r. The running time of the algorithm is given by the time to compute a maximum independent set in the largest neighborhood to be considered during the execution, which can be done in $n^{O(1/\varepsilon^2 \log 1/\varepsilon)}$.

Some extensions to different, related families of geometric intersection graphs are presented as well, including bounded and quasi disk graphs.

References

1. B.S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM*, 41(1):153–180, 1994.
2. H. Breu and D.G. Kirkpatrick. Unit disk graph recognition is NP-hard. *Computational Geometry. Theory and Applications*, 9(1-2):3–24, 1998.
3. T.M. Chan. Polynomial-time approximation schemes for packing and piercing fat objects. *Journal of Algorithms*, 46:178–189, 2003.
4. X. Cheng, X. Huang, D. Li, W. Wu, and D.-Z. Du. A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks. *Networks*, 42:202–208, 2003.
5. 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 (SODA'01)*, pages 671–679, Washington, DC, 2001.
6. R.L. Graham, D.E. Knuth, and O. Potashnik. *Concrete Mathematics*. Addison-Wesley, 2nd edition, 1998.
7. D.S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems. *Journal of the ACM*, 32(1):130–136, 1985.
8. 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.
9. F. Kuhn, R. Wattenhofer, and A. Zollinger. Ad-hoc networks beyond unit disk graphs. In *1st ACM DIALM-POMC Joint Workshop on Foundations of Distributed Computing*, San Diego, USA, 2003.
10. M.V. Marathe, H. Breu, H.B. Hunt III, S. S. Ravi, and D.J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25:59–68, 1995.

11. T. Nieberg, S. Dulman, P. Havinga, L.v. Hoessel, and J. Wu. Collaborative algorithms for communication in wireless sensor networks. In T. Basten, M. Geilen, and H. De Groot, editors, *Ambient Intelligence: Impact on Embedded System Design*. Kluwer Academic Publishers, 2003.
12. V. Raghavan and J. Spinrad. Robust algorithms for restricted domains. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 460–467. Society for Industrial and Applied Mathematics, 2001.

Appendix

Lemma 3. *For* $0 < \varepsilon < \frac{1}{10}$*, the value* $c = \frac{1}{\varepsilon^2} \ln \frac{1}{\varepsilon}$ *satisfies*

$$(2c+1)^2 < (3c)^2 < (1+\varepsilon)^c.$$

Proof. The first inequality is obviously true for the given choice of ε and c. The second inequality is equivalent to

$$2 \ln 3c < c \ln(1+\varepsilon),$$

(taking the log). This, again, is equivalent with

$$\frac{2}{c}(\ln 3 + \ln c) + \varepsilon^2 < \ln(1+\varepsilon) + \varepsilon^2.$$

In [6], it is shown that

$$\varepsilon \leq \ln(1+\varepsilon) + \varepsilon^2$$

holds for all $0 \leq \varepsilon < \frac{1}{2}$. Thus, it remains to show that $\frac{2}{c}(\ln 3 + \ln c) + \varepsilon^2 < \varepsilon$.

Note that, for $\varepsilon < \frac{1}{10}$, it is $\ln \frac{1}{\varepsilon} > 1$, and $\ln x < x$ for $x > 1$. Substituting $c = \frac{1}{\varepsilon^2} \ln \frac{1}{\varepsilon}$, we get

$$\begin{aligned}
\frac{2}{c}(\ln 3 + \ln c) + \varepsilon^2 &= \frac{2\varepsilon^2}{\ln \frac{1}{\varepsilon}}(\ln 3 + \ln(\frac{1}{\varepsilon^2} \ln \frac{1}{\varepsilon})) + \varepsilon^2 \\
&= 2\varepsilon^2(\frac{\ln 3}{\ln \frac{1}{\varepsilon}} + 2\frac{\ln \frac{1}{\varepsilon}}{\ln \frac{1}{\varepsilon}} + \frac{\ln \ln \frac{1}{\varepsilon}}{\ln \frac{1}{\varepsilon}} + \frac{1}{2}) \\
&< 2\varepsilon^2(\ln 3 + 2 + \frac{\ln \frac{1}{\varepsilon}}{\ln \frac{1}{\varepsilon}} + \frac{1}{2}) \\
&< 2\varepsilon^2(\ln 3 + 3.5).
\end{aligned}$$

Since $\varepsilon < \frac{1}{10}$, the inequality

$$2\varepsilon^2(\ln 3 + 3.5) < 10\varepsilon^2 < \varepsilon$$

holds and the claim follows. □