Unfolding MWIS on PTASGIG

Manassés Ferreira[1]

mfer@dcc.ufmg.br

[1]Dept. of Computer Science
Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte, Brazil

Introduction to Approximation Algorithms - 2014/2

Instructors: Olga Goussevskaia *and* Antonio Alfredo

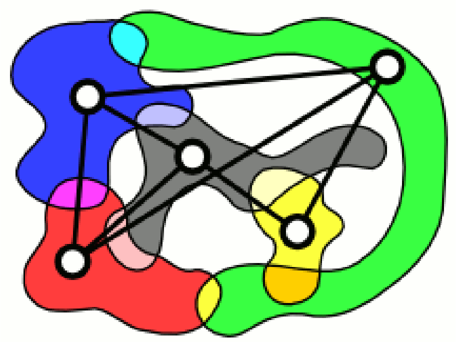
{olga, loureiro}@dcc.ufmg.br

Agenda

Polynomial-Time Algorithm Scheme for Geometric Intersection Graphs

$$n^{O(k^2)}$$

$$k > 1$$



$$\frac{1}{1+\epsilon} OPT_{IS}(\mathcal{D})$$

$$\epsilon > 0$$

Choosen Paper Cover

SIAM J. COMPUT.
Vol. 34, No. 6, pp. 1302–1323

POLYNOMIAL-TIME APPROXIMATION SCHEMES FOR GEOMETRIC INTERSECTION GRAPHS*

THOMAS ERLEBACH†, KLAUS JANSEN‡, AND EIKE SEIDEL‡

Abstract. A disk graph is the intersection graph of a set of disks with arbitrary diameters in the plane. For the case that the disk representation is given, we present polynomial-time approximation schemes (PTASs) for the maximum weight independent set problem (selecting disjoint disks of maximum total weight) and for the minimum weight vertex cover problem in disk graphs. These are the first known PTASs for $\mathcal{NP}$-hard optimization problems on disk graphs. They are based on a novel recursive subdivision of the plane that allows applying a shifting strategy on different levels simultaneously, so that a dynamic programming approach becomes feasible. The PTASs for disk graphs represent a common generalization of previous results for planar graphs and unit disk graphs. They can be extended to intersection graphs of other "disk-like" geometric objects (such as squares or regular polygons), also in higher dimensions.

Key words. independent set, vertex cover, shifting strategy, disk graph

AMS subject classifications. 68Q25, 68R10

DOI. 10.1137/S0097539702402676

Citations at Google Scholar

Polynomial-time approximation schemes for **geometric intersection graphs**

T Erlebach, K Jansen, E Seidel - SIAM Journal on Computing, 2005 - SIAM

A disk **graph** is the **intersection graph** of a set of disks with arbitrary diameters in the plane. For the case that the disk representation is given, we present **polynomial-time approximation schemes** (PTASs) for the maximum weight independent set problem (selecting disjoint **...**

Citado por 102 Artigos relacionados Todas as 5 versões Web of Science: 48 Citar Salvar

Applications

Automatic label placement, sometimes called **text placement** or **name placement**, comprises the computer methods of placing labels automatically on a map or chart. This is related to the typographic design of such labels.

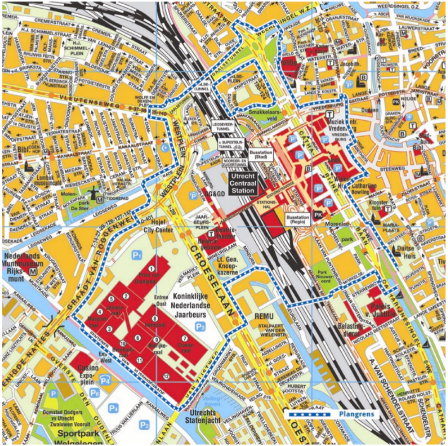
The typical features depicted on a geographic map are line features (e.g. roads), area features (countries, parcels, forests, lakes, etc.), and point features (villages, cities, etc.). In addition to depicting the map's features in a geographically accurate manner, it is of critical importance to place the names that identify these features, in a way that the reader knows instantly which name describes which feature.

Automatic text placement is one of the most difficult, complex, and time-consuming problems in mapmaking and GIS (Geographic Information System). Other kinds of computer-generated graphics – like charts, graphs etc. – require good placement of labels as well, not to mention engineering drawings, and professional programs which produce these drawings and charts, like spreadsheets (e.g. Microsoft Excel) or computational software programs (e.g. Mathematica).

Naively placed labels overlap excessively, resulting in a map that is difficult or even impossible to read. Therefore, a GIS must allow a few possible placements of each label, and often also an option of resizing, rotating, or even removing (suppressing) the label. Then, it selects a set of placements that results in the least overlap, and has other desirable properties. For all but the most trivial setups, the problem is NP-hard.

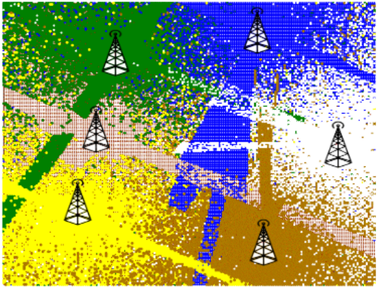
A **cellular network** or **mobile network** is a wireless network distributed over land areas called cells, each served by at least one fixed-location transceiver, known as a cell site or base station. In a cellular network, each cell uses a different set of frequencies from neighboring cells, to avoid interference and provide guaranteed bandwidth within each cell.

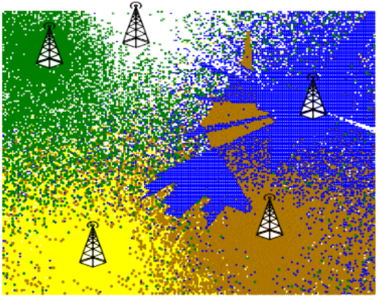
Automatic Label Placement



Given features on a map and the labels that belong to these features, place the labels near the features without labels overlapping other labels or overlapping features on the map

Cellular Network





The call control problem in a network that supports a spectrum of w available frequencies is to assign frequencies to users so that signal interference is avoided, while maximizing the number of users served.

Problems and Results

PTAS for MWIS and MWVC in the intersection graphs of disks, squares and other *disk-like* objects, also in higher dimensions.

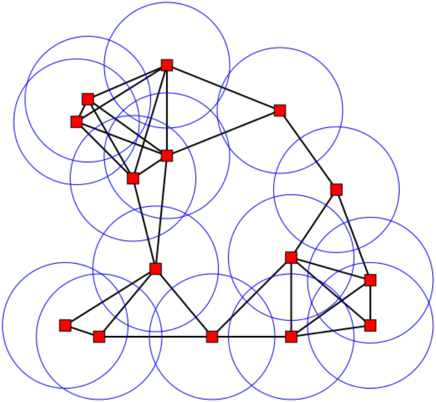
Maximum Weight Independet Set

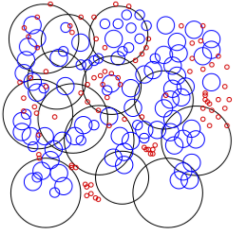
Compute, for a given set of geometric objects with certain weights, a subset of disjoint (**non-overlapping**) objects with maximum total weight.

Minimum Weight Vertex Cover

Compute asubset of the given objects with minimum total weight such that, for any two intersecting objects, at least one of the objects is contained in the subset.

For a set V of geometric objects, the corresponding intersection graph is the undirected graph with vertex set V and an edge between two vertices if the corresponding objects intersect.





Assume that we are given a set $\mathcal{D} = \{D_1, \ldots, D_n\}$ of n (topologically closed) disks in the plane, where D_i has diameter d_i, center $c_i = (x_i, y_i)$, and weight w_i. For a subset $U \subseteq \mathcal{D}$, $w(U)$ denotes the sum of the weights of the disks in U. Disks D_i and D_j *intersect* if $dist(c_i, c_j) \leq (d_i + d_j)/2$, where $dist(p_1, p_2)$ denotes the Euclidean distance between two points p_1 and p_2 in the plane. A *disk graph* is the intersection graph of a set of disks. We assume that the input to our algorithms is the set $\mathcal{D}$ of disks, not only the corresponding intersection graph. This is an important distinction, because determining for a given graph whether it is a disk graph is known to be $\mathcal{NP}$-hard [16], and hence no efficient method is known for computing a disk representation if only the intersection graph is given.

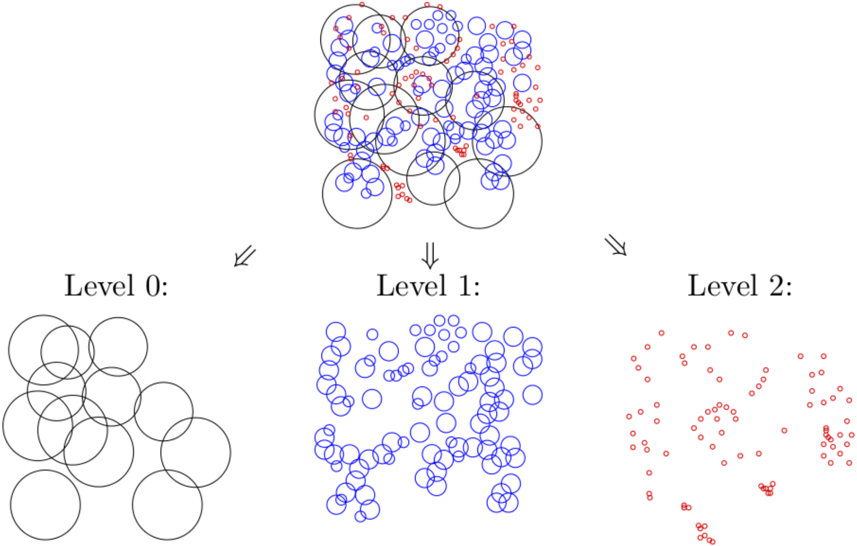
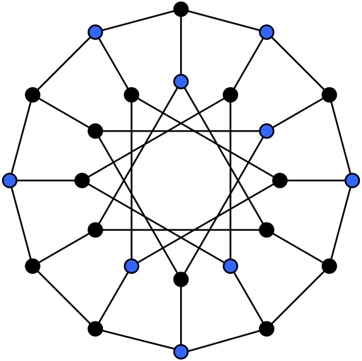


FIG. 2.1. *Partitioning the disks into levels* $(k = 2)$.

Maximum Weight Independet Set

Goal

Compute, for a given set of geometric objects with certain weights, a subset of disjoint (**non-overlapping**) objects with maximum total weight.



Generalized Petersen graph GP(12,4)

Divide and Conquer

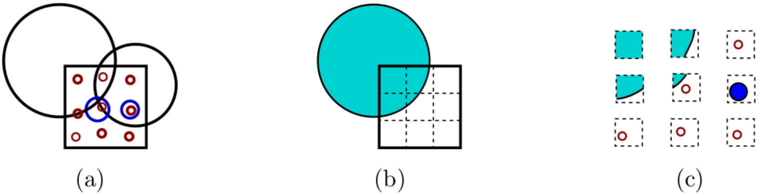


FIG. 2.5. *Example of table lookups for a square S at level j in case $k = 2$.* (a) *shows* 13 *disks in $\mathcal{D}(r, s)$ that intersect S:* 2 *disks of level less than j,* 2 *disks on level j, and* 9 *disks on level $j + 1$.* (b) *displays an independent set I consisting of* 1 *disk of level less than j.* (c) *illustrates that lookups are performed in* 9 *tables $T_{S'_{g,h}}$ during the computation of the table entries $AT_{S,I}(S'_{g,h}, *)$.*

Running-time

$$\underbrace{k^2 \cdot n^{O(1)}}_{relevantSquare} \cdot \underbrace{(\underbrace{O(k^2) \cdot n^{O(k^2)}}_{missingTable} + \underbrace{n^{O(k^2)}}_{enumerateSet} \cdot \underbrace{O(k^2) \cdot n^{O(k^2)}}_{computeAuxTable})}_{computeTable}$$

Procedures

relevantSquare Compute relevant squares and their forest structure

missingTable Compute missing tables $T_{S'_{g,h}}$

enumerateSet Enumerate sets I for which $T_S(I)$ has to be computed

computeAuxTable Compute $AT_{S,I}()$ for each $S'_{g_1..g_2,h_1..h_2}$

Dividing into Squares

Input: square S on level j,
set I of disjoint disks of level $< j$ intersecting S,
integers g, h with $0 \le g, h \le k$
Output: table entries $AT_{S,I}(S'_{g,h}, J)$ for all J
$AT_{S,I}(S'_{g,h}, *) \leftarrow$ undefined;
$Q \leftarrow$ all disks in $\mathcal{D}(r, s)$ of level j intersecting $S'_{g,h}$;
for all $U \subseteq Q$ such that $|U| \le Ck^2$ **do**
  if the disks in $I \cup U$ are disjoint **then**
    $I' \leftarrow \{D \in I \mid D$ intersects $S'_{g,h}\}$;
    $X \leftarrow T_{S'_{g,h}}(I' \cup U)$;
    $X \leftarrow X \cup \{D \in U \mid D$ is contained in $S'_{g,h}\}$;
    $J \leftarrow \{D \in U \mid D$ intersects the boundary of $S'_{g,h}\}$;
    if $AT_{S,I}(S'_{g,h}, J)$ is undefined **or**
      $w(X) > w(AT_{S,I}(S'_{g,h}, J))$ **then**
        $AT_{S,I}(S'_{g,h}, J) \leftarrow X$;
    fi
  fi
od

FIG. 2.4. *Computing the auxiliary table* $AT_{S,I}(S'_{g,h}, *)$.

Split

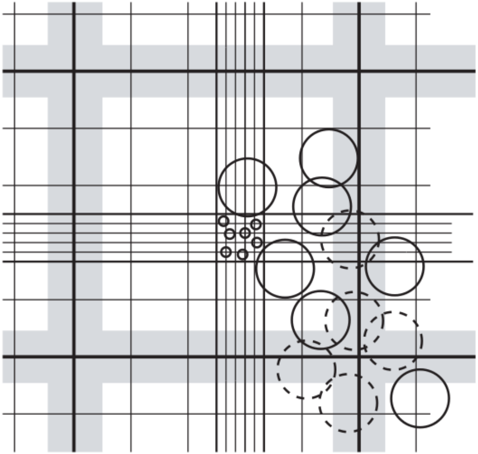


FIG. 2.3. *Example of grid and active lines on level j (coarse grid) and on level $j+1$ (fine grid) for $k = 5$. The big disks have level j, and the small disks have level $j+1$. All disks shown have the maximum possible diameter on their level. Active lines are drawn bold. Disks that hit active lines are drawn dashed. Note that a disk on level j can hit an active line only if its center is in the shaded strip along that active line.*

Combining Rectangles

Input: square S on level j,
set I of disjoint disks of level $< j$ intersecting S,
integers g_1, g_2, g_3 with $0 \leq g_1 \leq g_2 < g_3 \leq k$,
integers h_1, h_2 with $0 \leq h_1 \leq h_2 \leq k$,
previously computed table entries $AT_{S,I}(S'_{g_1 \cdot\cdot g_2, h_1 \cdot\cdot h_2}, *)$
and $AT_{S,I}(S'_{g_2+1 \cdot\cdot g_3, h_1 \cdot\cdot h_2}, *)$
Output: table entries $AT_{S,I}(S'_{g_1 \cdot\cdot g_3, h_1 \cdot\cdot h_2}, J)$ for all J
$R_1 \leftarrow S'_{g_1 \cdot\cdot g_2, h_1 \cdot\cdot h_2}$;
$R_2 \leftarrow S'_{g_2+1 \cdot\cdot g_3, h_1 \cdot\cdot h_2}$;
$AT_{S,I}(S'_{g_1 \cdot\cdot g_3, h_1 \cdot\cdot h_2}, *) \leftarrow$ undefined;
$Q \leftarrow$ all disks in $\mathcal{D}(r, s)$ of level j intersecting the boundary of R_1 or R_2;
for all $U \subseteq Q$ such that $|U| \leq 2C'k^2$ **do**
  if the disks in $I \cup U$ are disjoint **then**
    for $i = 1$ **to** 2 **do**
      $U_i \leftarrow \{D \in U \mid D$ intersects the boundary of $R_i\}$;
      $X_i \leftarrow AT_{S,I}(R_i, U_i)$;
    od;
    $X \leftarrow X_1 \cup X_2 \cup \{D \in U \mid D$ does not intersect the boundary of $S'_{g_1 \cdot\cdot g_3, h_1 \cdot\cdot h_2}\}$;
    $J \leftarrow \{D \in U \mid D$ intersects the boundary of $S'_{g_1 \cdot\cdot g_3, h_1 \cdot\cdot h_2}\}$;
    if $AT_{S,I}(S'_{g_1 \cdot\cdot g_3, h_1 \cdot\cdot h_2}, J)$ is undefined **or**
     $w(X) > w(AT_{S,I}(S'_{g_1 \cdot\cdot g_3, h_1 \cdot\cdot h_2}, J))$ **then**
      $AT_{S,I}(S'_{g_1 \cdot\cdot g_3, h_1 \cdot\cdot h_2}, J) \leftarrow X$;
    fi
  fi
od

FIG. 2.7. *Computing the auxiliary table* $AT_{S,I}(S'_{g_1 \cdot\cdot g_3, h_1 \cdot\cdot h_2}, *)$.

Merge

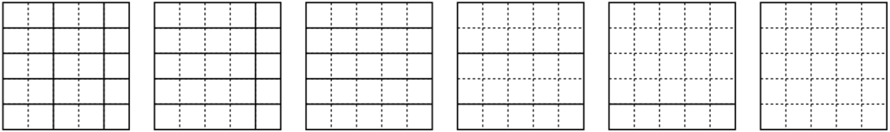


FIG. 2.6. *Combining subsquares into rectangles.*

Running-time

Polynomial for fixed $k > 1$

$$k^2 \cdot n^{O(1)} \cdot \left(O(k^2) \cdot n^{O(k^2)} + n^{O(k^2)} \cdot O(k^2) \cdot n^{O(k^2)} \right) = n^{O(k^2)}$$

Existence and Bound

Theorem 2.10

There is a PTAS for MWIS in disk graphs, provided that a disk representation of the graph is given. The running-time for achieving approximation ratio $1 + \epsilon$ is

$$n^{O\left(\frac{1}{\epsilon^2}\right)}$$

for a disk graph with n disks.

Proof

Lemma 2.1. *For at least one pair* (r, s), $0 \leq r, s < k$, *we have* $OPT(\mathcal{D}(r, s)) \geq (1 - \frac{1}{k})^2 OPT(\mathcal{D})$.

Proof. Let $S^* \subseteq \mathcal{D}$ be any set of disjoint disks with total weight $OPT(\mathcal{D})$.

For $0 \leq r < k$, let S^*_r be the set of all disks in S^* that hit a vertical line on their level whose index modulo k is r. As the sets S^*_r are disjoint, the weight of at least one of them must be at most a $\frac{1}{k}$-fraction of the weight of S^*. For this set S^*_r, let $T^* = S^* \setminus S^*_r$ and note that the weight of T^* is at least $(1 - \frac{1}{k})OPT(\mathcal{D})$.

For $0 \leq s < k$, let T^*_s be the set of all disks in T^* that hit a horizontal line on their level whose index modulo k is s. The weight of at least one of these sets T^*_s must be at most a $\frac{1}{k}$-fraction of the weight of T^*. For this set T^*_s, let $U^* = T^* \setminus T^*_s$. Note that $U^* \subseteq \mathcal{D}(r, s)$ and the weight of U^* is at least $(1 - \frac{1}{k})^2 OPT(\mathcal{D})$. □

Approximation Ratio ρ

The algorithm considers all k^2 possible values for r and s such that $0 \leq r, s < k$. For each possibility, an optimal independent set in $\mathcal{D}(r, s)$ is computed using dynamic programming. Among the k^2 sets obtained in this way, the one with largest weight is output. By Lemma 2.1, this set has total weight at least $(1 - \frac{1}{k})^2 OPT(\mathcal{D})$. Therefore, the algorithm achieves approximation ratio $(1 + \frac{1}{k-1})^2$. As k gets larger, the approximation ratio gets arbitrarily close to 1.

$$\rho = \left(1 + \frac{1}{k-1}\right)^2 \leq 1 + \frac{3}{k-1} = 1 + \epsilon$$

$$\epsilon = \frac{3}{k-1}$$

Running-time function of ϵ

$$\epsilon = \frac{3}{k-1} \Rightarrow k = \left\lceil \frac{3}{\epsilon} + 1 \right\rceil$$

$$n^{O(k^2)} \Rightarrow n^{O\left(\frac{1}{\epsilon^2}\right)}$$

Independent Set is not so hard on disk graphs and ...

... the PTAS can be generalized to d dimensions

There is a PTAS for MWIS in disk-like graphs, provided that a disk-like representation of the graph is given. The running-time for achieving approximation ratio $1+\epsilon$ is

$$n^{O\left(\frac{1}{\epsilon^{2(d-1)}}\right)}$$

for a disk-like graph with n disk-like objects and any constant d.

Questions