Approximation Algorithms for finding Independent Sets of Axis-Aligned Rectangles

Teena Carroll and Ye Yan

1 Introduction

A hotel has one conference room and many groups wishing to schedule meetings in that room during the week. They want to allow as many groups as possible to use the room given the time constraints of the meetings. We can represent each proposed meeting as an interval on the real line. Then selecting the maximum number of disjoint intervals is equivalent to scheduling as many meetings as possible so that our room is never overbooked.

It is easy to see there are many applications where we might want to consider "meetings" with two properties. This comes up in manufacturing processes, a product takes a certain amont of time to more through production, and certain machines must be used in a given sequence. We want to make as many products as possible so no machine is being used for two different products at the same time. This leads us to a two dimensional analog, representing each product as an axis aligned rectangle, with time on the y-axis and production sequence on the $x-axis$. Now our task is to find the highest number of rectangles so that no two rectangles intersect.

We can translate this into a graph theory problem. If we let every rectangle be a vertex and add edges between any vertices whose rectangles overlap, we get an *intersection graph*. Now, we want to find the largest independent set in this resulting graph.

Given any graph, it is difficult to find the size of the largest independent set. The best we can do is a polynomial approximation ratio of $n^{1-\epsilon}$ for any $\epsilon > 0$. For the class intersection graphs, the problem remains NP-hard, as shown in [4]. However, there are efficient approximation algorithms, and several subclasses of these graphs for which there approximation algorithms with low contant performance ratios.

Clearly these applications can be generalized to the weighted case, where each interval or rectangle is associated with a profit gained from choosing that event. It becomes our goal to maximize profit instead of the sheer number of events. We will also discuss the more general problem of finding a maximum weighted independent set (MWIS) of weighted rectangles.

2 Interval Graphs [1]

If we are interested in the one dimentional case, we can find the optimal independent set of a collection of intervals with a simple greedy algorithm. (Note that interval graphs are perfect, so we expect to be able to solve the independent set problem exactly and efficiently- definitions of perfection to follow).

We are given the collection of intervals by a list their two endpoints. $I_i = [a_i, b_i]$ for $i = 1 \dots n$. We sort the intervals in ascending order by their left endpoints. We take the interval with the highest left endpoint, call it l and add it to our independant set. Then we delete all intervals which intersect l. We repeat this procedure until no intervals remain.

This algorithm takes $O(n \log(n))$ time, as we have to sort the intervals by their endpoints, and all other steps are linear.

3 Unit Height Rectangles

On a map, we would like to be able to label regions, lines such as boundaries and rivers, as well as point data, like cities. There are potentially too many features of interest to be able to label all the features on a map with given size and resolution. We may also want the flexibility of using different fonts for different kinds of map elements. Each label has two parameters, length and height (as well as a fixed placement in the plane). Our goal is to label as many points as possible. We can see how an application of weighted rectangles may arise, as we would not want to leave the city of Atlanta unlabeled in order to label Decatur, Buckhead, and Dunwoody.

Agarwal and his co-authors provide an $O(n\log(n))$ 2-approximation algorithm in the case of unweighted unit height rectangles. In the map application, this translated into always using the same height font for all labels. They use a standard 'divide and conquer' approach to reduce this case to the one dimensional case. "Divide and Conquer" was the main technique used in this area until Mustafa's breakthrough paper [3].

Given a set R of unit- height rectangles, their algorithm first draws a set of horizontal lines, $l_1, l_2, \ldots, l_m$ with $m \leq n$, which satisfy

1. the distance between any two lines is strictly greater than one
2. each line intersects at least one rectangle
3. each rectangle is intersected by some line

Note that the first and third conditions together ensure that each rectangle is intersected by exactly one line. The lines partition the family of rectangles into subfamilies R_1, R_2, R_m, where the family R_i is the set of all rectangles intersecting line l_i.

Now by looking at the projections of the rectangles onto these lines as intervals, we can use the one dimentional greedy algorithm to find the optimal independent set, M_i, for each R_i in $O(|R_i|\log(|R_i|))$ time.

Because of the spacing of the lines, rectangles in R_i can only intersect rectangles from R_{i-1} and R_{i+1}. Therefore, if we look at $\bigcup_{\{i \text{ even}\}} M_i$ and $\bigcup_{\{i \text{ odd}\}} M_i$, both of these form independent sets in the original graph. By the pigeonhole principle, one of these sets has size at least half of the optimal.

The running time is $O(n\log(n))$, since initially we need to sort the rectangles by their y-coordinates, and then the lines can be assigned with a single pass through all the rectangles.

4 Axis Aligned Piercing Rectangles [3]

Given a set of rectangles, we can form a graph by representing each rectangle with a vertex, and adding an edge between any two rectangles that intersect.

Two rectangles r_1 and r_2 are said to be piercing if r_1 intersects both vertical edges of r_2 and r_2 intersects both horizontal edges of r_1. We use the notation that $r_1 \prec r_2$.

Notice that if $r_1 \prec r_2$, that means the two rectangles intersect and no one rectangle contains any vertex of the other. Also, $\prec$ is a transitive relationship, meaning that if $r_1 \prec r_2 \prec r_3$, then $r_1 \prec r_3$.

Given a set of rectangles, we can form the piercing intersection graph, G_p (which is a directed subgraph of the intersection graph) by including a directed edge from r_1 to r_2 iff $r_1 \prec r_2$.

Lemma 1 *Given a set R of n, an optimal independent set of G_p can be found in polynomial time*

A graph is *vertex transitive* if the automorphism group of the graph is transitive on its vertex-set.

This definition relies on a fair bit of algebra background to understand, however, all it is saying is that there are graph automorphisms (maps from the graph to itself which don't change the edge relationships between the vertices), which can send any vertex to any other vertex. Another way to say this is that the orbit of any vertex is the whole graph.

G_p is always a vertex transitive graph because it is defined based on a transitive relation.

A Graph is *perfect* if for the graph and all its induced subgraphs, the coloring number equals the clique number.

The clique number (the size of the largest induced complete subgraph) is a lower bound for how many colors we need to color the vertex set of the graph so no two vertices of the same color are connected. A perfect graph can be colored with this number of colors.

It is a well-known fact that vertex-transitive graphs are perfect.

Though the problem of finding large independent sets is in general hard, perfect graphs are a class of graphs which are known to have polynomially computable maximum independent sets. We will not prove that here, but a polynomial time algorithm using the ellipsoid method was developed by Grotschel, Lovasz, and Schrijver.

5 Axis Aligned Non-Piercing Rectangles

Let R be a set of non-piercing rectangles. Computing the optimal independent set of in the class of intersection graphs of non-piercing rectangles remains NP-hard [1].

Let $I^*(R)$ be a maximum independent set in G_R. So, $|I^*(R)|$ is the size of an optimal solution. We can approximate this solution by a constant factor of 11 by finding an "r-maximal" set, an appropriately large indepdent set which we can find in polynomial time.

We call a subset $R' = r'_1, ...r'_m \subseteq R$ r-maximal if it satisfies these 4 conditions:

1. R' is an independent set
2. Every $r \in R \setminus R'$ intersects some $r' \in R'$
3. For each $r' \in R'$, there is at most one rectangle $r \in I^*(R)$ such that r intersects r' and does not intersect any other rectangle in R'
4. For every pair of rectangles $r', t' \in R'$, there are at most two rectangles $r, t \in R \setminus R'$ such that $r \cap t = \emptyset$, and r, t intersect both r' and t' and no other rectangles in R'.

Lemma 2 *Let $R' = r'_i ... r'_m \subseteq R$ be an r-maximal set. Then $|R'| \geq \alpha(R)/c_0$, where $c_0 \leq 11$.*

We defined the r-maximal set such that some rectangles in the maximum independent set of R can be charged to the rectangles in the r-maximal set. Let $I = I^*(R)$ be the set of rectangles in the maximum independent set, we charge a rectangle $r \in I$ to a rectangle in R' depending on how they intersect:

1. r does not intersect any rectangle in R'. Not possible due to the maximality condition (condition 2 from above).
2. r pierces a rectangle in R'. Not possible since R contains only non-piercing rectangles.

3. r only intersects one rectangle $r' \in R'$. Charge r to r'. By condition 3 from above, each rectangle in R' receives at most one such charge.

4. r intersects r' in one or more corners (i.e., r contains one or more vertices of r'). Charge r to r'. Since I and R' are both independent sets, this ensures that each corner of r' is in at most one r. So each element of R' receives at most 4 such charges.

Because of the way we defined r-maximal, we're guaranteed that for each rectangle in the r-maximal set, at most 5 rectangles in the maximum independent set can be charged to it.

Now we need to put a upper bound on the number of uncharged rectangles in I. Let $I' \subseteq I$ be the uncharged rectangles. Each rectangle in I' intersects exactly two rectangles in R' (by r-maximal property 4 and the removal of all rectangles which intersect exactly one).

Now we can make a graph $G_{I'}$ by mapping each rectangle in R' to a vertex and each rectangle in I' to an edge, so there is an edge between two vertices of there is a rectangle in I' which intersects both of their respective rectangles.

Lemma 3 *$G_{I'}$ is a planar multi-graph.*

Formal proof omitted (draw pictures to convince yourself).

For planar graphs, the number of edges is at most three times the number of vertices. Thus $|I'| = |E| \leq 6|V| = 6|R'|$.

Thus, the number of uncharged rectangles in the maximum independent set is guaranteed to be at most 6 times the number of rectangles in the r-maximal set. $5 + 6 = 11$, so we can approximate the maximum independent set by a factor of 11 using the r-maximal set.

6 Weighted Case [2]

It is clear why weighted rectangles are needed in many applications where an axis aligned rectangle model is relevant. We want not to make the highest number of products, but we want to make the set of products which will maximize our profit, even if the size of our product line is smaller.

Lewin-Eytan, Naor and Orda describe an algorithm which approximates the optimal profit by a factor of $4c$, where c is the maximum clique size of the intersection graph of the rectangles. Another way to define c' is that it is the maximum number of rectangles which cover any given point in the plane.

Given a set R of rectangles $(r_1....r_n)$, with weights $w(r_i) \geq 0$, for each independant set, we assign indicator variables x_i, so $x_i = 1$ iff r_i is included in a feasible solution, it is an integer program to maximize the weight of the chosen rectangles, i.e. we want to find an independent set I with maximal

$$\sum_{r \in I} w(r) \cdot x_r$$

Then for each clique Q, we want the constraint that:

$$\sum_{v \in Q} x(v) \leq 1]$$

They relax this into a linear program, and introduce a rounding technique which yields an integer solution which does no worse than 4 times the optimal solution. The solution may not be an independent set however. All we are guarenteed is that the set returned from the rounding process has the property that no rectangles are *vertex incident*: none of the rectangles contain the corner of another. Let's call the solution returned S.

Theorem 1 *The incidence graph for a set of non-vertex incident rectangles is perfect*

Let c' be the clique size of the graph induced by S. We have the $c' \leq c$ since $S \subseteq R$. Since the graph on S is perfect we can color it with exactly c' colors. We can look at each color class, and pick the one with the highest profit. By pigeonhole principle, that one will have at least $\frac{1}{c}$ of the total weight of S. This gives us at least $\frac{1}{4c}$ of the total weight of the best solution from all rectangles in R.

References

[1] Agarwal, Krevald, Suri *Label Placement by maximuum independent set in Rectangles.* Computational Geometry Theory and Applications vol 11 p209-218 (1998)

[2] Lewin-Eytan, Naor and Orda *Routing and Admission Control in Networks with Advance Reservations.* APPROX p215-228 (2002)

[3] Mustafa *Approximation Algorithms for Independent set of Rectangles when the Independent set is Large.* Manuscript (2004).

[4] Rim and Nakajima *On Rectangle intersection and overlap graphs.* IEEE Transactions on Circuits and Systems vol 9 p549-553 (1995)