Algorithmic Modelling and Complexity **Fall 2003**

Lecture 3: 9 September

Lecturer: J. van Leeuwen *Scribes: Shay Uzery & Kasper van den Berg*

3.1 Overview

During this lecture we will consider *facility location problems*. A typical question in facility location is to determine a set of locations in a network such as all other locations (nodes) are within "easy reach". Facility location problems can be modelled by network models where the edges are labelled with 'distances' or 'costs'. In this lecture we will discuss the following facility location problems: 'finding dominating locations', the 'k-center problem', and the 'lazy tourist problem'.

3.2 Finding dominating locations

Finding dominating locations (i.e. finding a dominating set) is a basic case of the facility location problem. Here only existing edges count, and their weight is ignored.

Definition 3.1 *Let $G = \langle V, E\rangle$ be an unweighted network. $S \subseteq V$ is called a dominating set in G if for every $u \in V$ either $u \in S$ or u is adjacent to a node s in S, i.e. $(u, s) \in E$.*

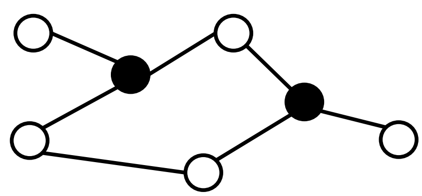

Figure 1: The black nodes form a dominating set in this graph

Finding a minimum size dominating set is computationally hard – there is no known polynomial time algorithm to compute it. We assume from now on that the networks we consider are connected. If a network consists of unconnected subgraphs, the minimum dominating set is the union of the minimum dominating sets of the subgraphs.

The 'naïve' way to compute it is to check all $\binom{n}{k}$ subsets $\subseteq V$ of size k, and incrementing k from 1 to n in steps until a minimum size dominating set is found. Let OPT be the size of a minimum size dominating set in G.

Proposition 3.2 *A minimum dominating set can be found in a number of steps bounded by $O(\log OPT \cdot \max_{0\le k\le 2\cdot OPT}\binom{n}{k})$.*

Proof: If there is a dominating set of size OPT then there is one of any size larger than OPT (any superset will do). A minimum size set can thus be found as follows. Increment k in powers of two $(1, 2, 4, \dots)$ until a j is found such that there is *no* dominating set of size 2^{j-1} but there is one of size 2^j or n, whichever is smaller. Carry out a binary search between 2^{j-1} and 2^j to find the smallest k for which a dominating set of size k exists. The search takes $O(j) = O(\log OPT)$ iterations, and each iteration involves testing $\binom{n}{k}$ subsets, for some $k \leq 2 \cdot OPT$. ■

Can one bound OPT? Let D_G be the minimum degree of any vertex in G.

Lemma 3.3 (Alon and Spencer, 1992) $OPT \leq n\frac{1+\ln(D_G+1)}{D_G+1}$.

Proof: We will show that $n\frac{1+\ln(D_G+1)}{D_G+1}$ is an upper bound to the size of a dominating set in a graph with a minimum degree D_G (cf. [1]). We choose the vertices for the dominating set one by one, where in each step a vertex that covers the maximum number of yet uncovered vertices is chosen. Let $N(v)$ be the set consisting of v together with all its neighbours. Suppose that during the process of picking vertices, the number of vertices u that do not lie in the union of the sets $N(v)$ of the vertices chosen so far, is r. Remembering the assumption that the minimum degree is D_G, the sum of the cardinalities of the sets $N(u)$ over all uncovered vertices u is at least $r(D_G + 1)$, and by averaging, there is a vertex v that belongs to at least $\frac{r(D_G+1)}{n}$ such sets $N(u)$. Adding this v to the set of chosen vertices we observe that the number of uncovered vertices is now at most $r(1 - \frac{D_G+1}{n})$. It follows that in each iteration of the above procedure the number of uncovered vertices decreases by a factor of $1 - \frac{D_G+1}{n}$. Hence after $n\frac{\ln(D_G+1)}{D_G+1}$ steps there will be at most $\frac{n}{D_G+1}$ yet uncovered vertices, which can now be added to the set to form a dominating set of size $\leq n\frac{1+\ln(D_G+1)}{D_G+1}$. ■

Proposition 3.4 *A minimum dominating set can be found in a number of steps bounded by* $O(\log L \cdot \max_{0\leq k\leq L} \binom{n}{k})$*, with* $L = n\frac{1+\ln(D_G+1)}{D_G+1}$.

Proof: Use binary search in the range $[1 \dots n\frac{1+\ln(D_G+1)}{D_G+1}]$. The complexity of a binary search is $O(\log(search_range))$. Now argue as before. ■

In the general case of the dominating set problem, the best polynomial-time approximation algorithm known achieves a performance ratio of $1 + \log n$. It is a major open problem whether the dominating set problem could be solved in polynomial time by an algorithm with a performance ratio $\leq c \log n$ for some (small) $c < 1$.

There are special cases in which better than this can be done.

Theorem 3.5 (Baker, 1994) *For the planar network case, there is an algorithm A running on instances (G, ϵ) such that for any given $\epsilon > 0$, A produces a dominating set of size $\leq (1 + \epsilon)OPT$ in time complexity polynomial to $|G|$.*

The algorithm $\mathcal{A}$ may be exponential in $\frac{1}{\epsilon}$ however.

3.2.1 Upper- and lowerbounding minimum dominating sets

Let $G = \langle V, E\rangle$ be an unweighted (connected) network.

Definition 3.6 *$I \subseteq V$ is an independent set if every two nodes in I are not adjacent.*

Definition 3.7 *An independent set I is called maximal if for every node $v \notin I$, $I \cup \{v\}$ is not independent.*

Definition 3.8 *An independent set is called a maximum independent set if it has the largest possible size of any independent set in G.*

Computing a maximal independent set is 'easy'. It can be done in polynomial time as follows: start with $S = \emptyset$, add any $v \in V$ such that $\{v\} \cup S$ is an independent set, and iterate. The algorithm stops when for every $v \in V/_S$, $S \cup \{v\}$ is no longer independent.

Definition 3.9 *The square of a network $G = \langle V, E\rangle$ is the network $G^2 = \langle V', E'\rangle$, where $V' = V$ and $E' = E \cup \{(u, v) | u \neq v$ and u and v are within distance ≤ 2 in $G\}$.*

Note that squaring a planar network doesn't have to result in a planar network again.

Lemma 3.10 *Let I a maximal independent set in G, S any minimum dominating set in G, and J a maximal independent set in G^2. Then:*

$$|J| \leq |S| \leq |I| \tag{1}$$

Proof:
(a) $|S| \leq |I|$.

We will show that I is a dominating set in G. For any node $x \in V$ either $x \in I$ or x is within a distance 1 from a node in I, otherwise we could add x to I and $I \cup \{x\}$ would also be an independent set, which is a contradiction to I being a maximal independent set. Thus I is a dominating set in G. Since S is a minimum dominating set in G, $|S| \leq |I|$.

(b) $|J| \leq |S|$.

By squaring the network G, the neighbourhoods of all elements of S are turned into 'cliques' – groups of elements that are mutually connected. The cliques of the nodes $u \in S$ span the entire set V (some neighbourhoods of two nodes $u_1 \in S$ and $u_2 \in S$ might be combined into one clique but this does not matter). At most one node of each clique can be in J. Thus $|J| \leq |S|$. ■

3.3 k-Center problem

The next problem concerns a metric-symmetric network. Here we assume that *all* edges are present and weighted with 'distances' such that the triangle inequality is satisfied. We want to find a dominating set of size $\leq k$ such that the distances of all other nodes to one of the set nodes is minimised over all possible choices of dominating set of size $\leq k$.

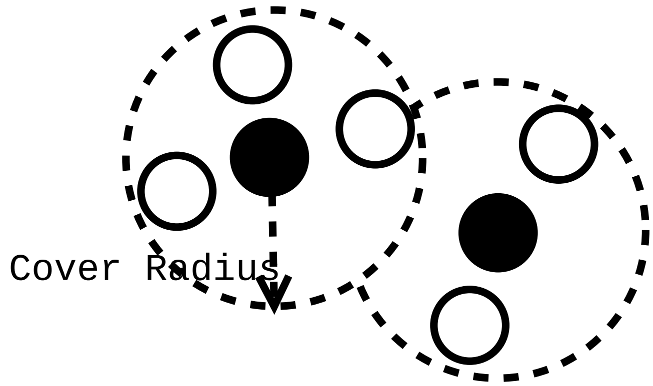

Figure 2: The black nodes are the centers which cover all the nodes in the graph

Definition 3.11 *The covering radius is the maximum of the minimum distances from any node not in the dominating set to a node that is in the dominating set.*

We will show the following result.

Theorem 3.12 (Hochbaum & Shmoys, 1985) *The k-center problem can be solved by an approximation algorithm in polynomial time with performance ratio ≤ 2.*

The key to the algorithm is a 'bottleneck' technique: start with an 'empty' network, add the edges of E one by one, and see what 'covering ranges' are created.

Algorithm HS

Step 1. Order the edges of G in *non-decreasing* order by length: $e_1, e_2, \ldots$ (thus $cost(e_1) \leq cost(e_2) \leq \ldots$).

(to be continued)

Definition 3.13 *Let G_i be the network with vertex set V and edges $e_1, \ldots, e_i$.*

Observation:

The k-center problem is now the problem of determining the smallest i such that G_i has a dominating set of size $\leq k$. The covering radius will be $cost(e_i)$.

Exercise. Argue that 'the smallest i' in the observation is well-defined. (Hint: what size dominating set does G_n have.)

Consider G_i for increasing i. The minimum dominating set in G_i will get smaller and smaller as i increases. Lemma 3.10 gives a lowerbound $|J|$ on its size which is easy to compute, taking J to be any maximal independent set in G_i^2. $|J|$ decreases also when more edges are added. As long as $|J| > k$, G_i cannot have a dominating set of size $< k$. We exploit this as follows.

Algorithm HS (continued)

Step 2. Start with a network $G_0 = \langle V, E_0 \rangle$ containing all vertices and no edges. Set i=1.

Step 3. Add edge e_i to obtain the network G_i.

Step 4. Square G_i resulting in G_i^2.

Step 5. Compute a maximal independent set M_i in G_i^2.

Step 6. If $|M_i| > k$, then $i := i + 1$ and go back to step 3.

Step 7. Return M_i.
Comment: At this point i is the first index such that $|M_i| \leq k$.

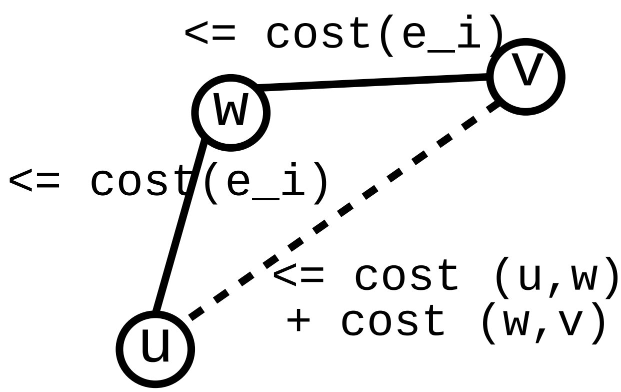

Figure 3: Node u is always within a distance of $2 \cdot cost(e_i)$ from a node $v \in M_i$

Theorem 3.14 *Algorithm HS always returns a solution to the k-center problem with a covering radius* $\leq 2 \cdot OPT$.

Proof: Let G_j be the first network with a solution to the problem. According to lemma 3.10 we get $1 \leq i \leq j$. This results in $G_1 \ldots G_i \ldots G_j$ and hence $cost(e_i) \leq OPT$. We now consider the quality of the set M_i returned by the algorithm.

Clearly $|M_i| \leq k$, thus M_i is a feasible solution. Consider any node $u \in G$. M_i is a maximal independent set in G_i^2. From this follows that M_i is also a dominating set in G_i^2. Thus u is either $\in M_i$ or is connected by an edge to some node $v \in M_i$ (in G_i^2). But then u is within a distance of two edges from v in G_i and both edges have cost $\leq cost(e_i)$ (see illustration 3.3). According to the metric property of the network it is concluded that the cost from u to v is $\leq 2cost(e_i) \leq 2 \cdot OPT$. ■

We note that algorithm HS is easily implemented to run in low polynomial time in $|G|$.

3.3.1 Yet another approach to the k-center problem

Algorithm HS does not build a candidate solution set for k-center problem incrementally. A *greedy approach* does. Let $G = \langle V, E \rangle$ be a metric-symmetric network again.

Definition 3.15 *For $u \in V$ and any subset $J \subseteq V$ let $d(u, J)$ be the shortest distance of u to any node in J.*

Consider the following algorithm that builds up a candidate set J in stages. In every stage it adds the 'farthest' node into the set.

Algorithm G

Pick any initial vertex v_0. Set $J_0 = \{0\}$.

for $i = 1$ **to** k **do**

begin
$v_i :=$ any node farthest from J_{i-1}
$d_i :=$ the distance of $v_i :=$ to J_{i-1}
$J_i := J_{i-1} \cup \{v_i\}$
end

return $J = J_{k-1}$

comment: v_k is not included in the returned set

comment: v_k is farthest from J_{k-1}, thus d_k is the covering radius of $J = J_{k-1}$

Theorem 3.16 (Gonzalez, 1985) *Algorithm G always returns a solution to the k-center problem with a covering radius $\leq 2 \cdot OPT$.*

Proof: Clearly $|J| \leq k$ by construction. We now consider the quality of the set $J = J_{k-1}$.

Let C be an optimum solution and let its covering radius be OPT. As d_k is the covering radius of J it follows that $OPT \leq d_k$.

To measure the distance of any node to J it suffices to estimate d_k, as v_k is at farthest distance from it. Consider $J_k = \{v_0, \ldots, v_k\}$, a set with $k+1$ elements. C covers the network with $\leq k$ 'circles' of radius $\leq OPT$. Thus there must be a $u \in C$ such that the circle around u contains 2 elements v_i and v_j, with necessarily $d(v_i, v_j) \leq OPT$. Say $i < j$, thus v_j is added *later* to J than v_i and $v_i \in J_{j-1}$. Now observe

$$d_k = d(v_k, J_{k-1}) \leq d(v_j, J_{j-1}) \leq d(v_j, v_i) \leq d(v_j, c) + d(c, v_i) \leq 2 \cdot OPT.$$

The first inequality follows because farthest distances decrease as j increases (or note that $d(v_k, J_{k-1}) \leq d(v_k, J_{j-1}) \leq d(v_j, J_{j-1})$ by the choice of v_j in the jth round). We conclude that d_k is within twice OPT. ∎

Algorithm G again has an easy polynomial running time.

3.3.2 Can the approximation ratio be improved

Both polynomial time algorithms we saw for the k-center problem achieve a performance ratio of 2. Doing better seems to be very difficult. In fact, if it could be done we would have solved a major open problem as stated in the following observation.

Theorem 3.17 *If the metric, symmetric k-center problem could be solved by a polynomial time algorithm with a performance ratio < 2 than the general dominating set problem could be solved in a polynomial time.*

Proof: The dominating set problem in the 'decision' variant: given k, is there a dominating set of size k. We reduce the dominating set problem to an instance of the k-center problem.

Let $G = \langle V, E \rangle$ be any network, k the parameter for the dominating set query. Design a 'complete' symmetric network G' with the following costs:

- $cost(u, v) = 1$ if (u, v) is an edge in G.
- $cost(u, v) = 2$ if (u, v) is not an edge in G.

It is easily seen that G' is metric. There is a k-center with radius ≤ 1 in G' iff there is a dominating set of size $\leq k$ in G.

Suppose the metric, symmetric k-center problem could be solved by a polynomial time algorithm $\mathcal{A}$ with a performance ratio < 2. Run $\mathcal{A}$ on G' for $k = 1$. It will return an approximation in the range $[1, 2\rangle$ if there is a solution. It will return a value ≥ 2 if there is no solution. Thus, the approximation algorithm can be used to decide the dominating set problem in polynomial time. ■

Solving the dominating set problem in polynomial time is intimately connected with the P-versus-NP problem. By the theorem so is the problem of finding a polynomial time algorithm for the k-center problem with a performance ratio better that 2!

Variants of the k-center problem continue to be studied. It is a special case of the *k-supplier problem*: given a set of suppliers Σ and a set of customers C, determine a subset $S \subseteq \Sigma$ with $|S| \leq k$ that covers C with smallest possible covering radius.

3.4 The lazy tourist problem

Consider a lazy tourist in a big city. Model the area he wants to visit as a graph, with the nodes being all sites that can be visited and edges (u, v) representing that site u is visible from v and vice versa.

The tourist wants to make a closed walk visiting the fewest possible number of places while seeing *all* of them.[1]

[1] A walk is a tour that may pass through a node more than once.

The lazy tourist problem can be approximated by finding a minimun size connected dominating set and vice versa - a lazy lazy tourist walk constitutes a connected dominating set [3].

Proposition 3.18 (a) *If there exists a lazy tourist walk in G with k edges, then G has a connected dominating set of size $\leq k$.*

(b) *If G has a connected dominating set of size $\leq k$, then there is a lazy tourist walk in G with $\leq 2 \cdot k$ edges.*

Proof:

(a) Take the $\leq k$ vertices visited during the walk.

b Let M be a connected dominating set of size $\leq k$, T a spanning tree of M. Do a depth-first traversal of T. ■

The proposition implies that whenever we have a polynomial time approximation algorithm for the connected dominating set problem with performace ratio α, then we automatically have a polynomial time approximation for the lazy tourist problem with a performance ratio $\leq 2 \cdot \alpha$.

Theorem 3.19 (Guha and Khuller, 2001) *There is a polynomial time algorithm that computes a connected dominating set within a performance ratio of $\ln \Delta + 3$, where Δ is the maximum degree in G. In case vertices have weights and a minimum weight connected dominating set is sought, a polynomial time approximation algorithm exists that achieves a performace ratio of $3 \cdot \ln n$.*

References

[1] N. Alon, J.H. Spencer. *The probabilistic method.* J. Wiley & Sons, new York, 1992.

[2] B.S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM* 41:1 (1994) 153-180.

[3] S. Guha, S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica* 20:4 (1998) 374-387.

[4] D.S. Hochbaum, D.B. Shmoys. A best possible heuristic for the k-center problem. *Math. of Operations Research* 10 (1985) 180-184.