MATHEMATICS OF OPERATIONS RESEARCH
Vol. 10, No. 2, May 1985
Printed in U.S.A.

A BEST POSSIBLE HEURISTIC FOR THE k-CENTER PROBLEM*

DORIT S. HOCHBAUM† AND DAVID B. SHMOYS‡

University of California, Berkeley

In this paper we present a 2-approximation algorithm for the k-center problem with triangle inequality. This result is "best possible" since for any $\delta < 2$ the existence of δ-approximation algorithm would imply that $P = NP$. It should be noted that no δ-approximation algorithm, for any constant δ, has been reported to date. Linear programming duality theory provides interesting insight to the problem and enables us to derive, in $O(|E|\log|E|)$ time, a solution with value no more than twice the k-center optimal value.

A by-product of the analysis is an $O(|E|)$ algorithm that identifies a dominating set in G^2, the square of a graph G, the size of which is no larger than the size of the minimum dominating set in the graph G. The key combinatorial object used is called a strong stable set, and we prove the NP-completeness of the corresponding decision problem.

1. Introduction. An instance of the k-center problem consists of a complete graph $G = (V, E)$ with edge weights $w_e \geqslant 0$, $e \in E$ and $w_{(v,v)} = 0$, $v \in V$. The problem is to find a subset $S \subseteq V$ of size at most k such that $w(S) = \max_{i \in V} \min_{j \in S} w_{(i,j)}$ is minimized. In this paper we consider instances of the k-center problem that satisfy the triangle inequality, i.e. for every triple, $i, j, k \in V$, $w_{(i,j)} + w_{(j,k)} \geqslant w_{(i,k)}$.

A δ-approximation to the k-center problem is the problem of finding a set $S' \subseteq V$ of size k at most such that $w(S')$ is at most δ times the value of an optimal solution. The k-center problem with triangle inequality is not only NP-complete but also any δ-approximation for $\delta < 2$ is NP-hard ([H1], [HN]).

The k-center problem is intimately related to another problem—the dominating set problem (DS). An instance of DS is a graph $G = (V, E)$, and the problem is to identify a set $S \subseteq V$ of minimum size such that for all $v \in V - S$ there exists $s \in S$ with $(v, s) \in E$. The relationship between the two problems has been described and used in [H2]. We shall use precisely this reduction for our worst case analysis. The reduction relies on the fact that the k-center solution must assume one of $m = |E|$ values, $w_{e_1}, \ldots, w_{e_m}$. Let $W = w_{e_i}$ for some i. We define the W-graph of G, $G(W)$ to be $G(W) = (V, E_W)$ where $e \in E_W$ if and only if $w_e \leqslant W$. It can be easily verified that finding the solution to the k-center problem is equivalent to finding a minimum value of W such that the graph $G(W)$ has a dominating set of size not exceeding k. We call this dominating set S^*, and the value of the corresponding k-center optimal solution is $w(S^*)$. We call the graph corresponding to that minimum value of W the bottleneck graph, $G_B = G(w(S^*))$. Unfortunately, the problem DS is itself NP-complete and in that sense this reduction is not helpful.

We now define the square of a graph $G = (V, E)$ to be the graph $G^2 = (V, E^2)$ where $e \in E^2$ if and only if $e = (u, v) \in E$ or $\exists t \in V$ such that $(u, t) \in E$ and $(t, v) \in E$ (alternatively, if the shortest path between u and v in the graph G contains at most 2 edges).

*Received June 13, 1983; revised December 9, 1983.
AMS 1980 subject classification. Primary 05C99.
OR/MS Index 1978 subject classification. Primary: 485 Networks/graphs/heuristics.
Key words. k-center problem, dominating set, strong stable set.
†Supported in part by the National Science Foundation under grant ECS-8204695.
‡Supported in part by the National Science Foundation by a graduate fellowship and under grant MCS-7820054.

0364-765X/85/1002/0180$01.25

The square of a graph played a fundamental role in a recent result for a 2-approximation algorithm for the bottleneck TSP with triangle inequality [PR]. That result relied on the fact that a Hamiltonian circuit in the square of a biconnected graph can be found in polynomial time. The analogue in our case would be to solve the problem of finding a k-dominating set in the square of a graph. However, we showed that this problem is NP-complete. Our approach is based on the following idea. Rather than finding the optimal dominating set for the square of the graph, we will show that it suffices to identify a *feasible* dominating set that satisfies certain additional properties.

The following notation will be useful in the remainder of this paper. Consider a graph $G = (V, E)$. Let $N_G(u)$ denote the neighborhood of u, that is,

$$N_G(u) = \{ v \in V \mid (u, v) \in E \} \cup \{ u \}.$$

2. Squared graphs and bottleneck graphs. We begin by stating some simple, but useful facts about the squares of graphs, and bottleneck graphs.

LEMMA 1. *Consider the graph $G(W) = (V, E_W)$ corresponding to a complete edge weighted graph $G = (V, E)$ and a specified value W. Let $G(W)^2 = (V, E_W^2)$ be the square of the graph $G(W)$. If $e \in E_W^2$ then $e \in E_{2W}$, where $G(2W) = (V, E_{2W})$.*

PROOF. To prove the lemma we can equivalently show that for all $e \in E_W^2$, $w_e \leqslant 2W$. This follows directly from the fact that the weights satisfy the triangle inequality. ∎

Let S_W be a dominating set of size k in the graph $G(W)$ (if such a dominating set exists). Note that the value of the k-center solution corresponding to the set of centers S_W is at most W.

LEMMA 2. *Let $W \leqslant w(S^*)$. Consider S such that $|S| \leqslant k$ and S is dominating in the graph $G(W)^2$; then S is a feasible k-center solution with cost $w(S) \leqslant 2W \leqslant 2w(S^*)$.*

PROOF. This follows immediately from Lemma 1 and the observation made above. ∎

Consider G^2, the square of the graph $G = (V, E)$. We will show in §5 that the dominating set problem for G^2 is NP-complete. Let S^{2*} be a minimum dominating set for G^2. Since every dominating set of G is a dominating set for G^2, it follows that $|S^{2*}| \leqslant |S^*|$. Our strategy will be to identify a set S which is dominating in G^2 such that

$$|S^{2*}| \leqslant |S| \leqslant |S^*|. \tag{*}$$

3. The strong stable set problem. The key notion in finding the approximate k-center is that of a strong stable set. A subset of vertices S is a *strong stable set* if for each $u \in V$, $|N_G(u) \cap S| \leqslant 1$. In other words, a strong stable set S is a stable set (or independent set) with the additional restriction that every vertex not in S can be adjacent to at most one vertex in S. It is not hard to see that the problem of finding the largest strong stable set is dual to that of finding the smallest dominating set, in the usual linear programming sense of duality. We note that the problem of finding the largest strong stable set is NP-hard.

From the weak duality theorem of linear programming we derive the following lemma.

LEMMA 3. *Consider a graph $G = (V, E)$. Let $SS \subseteq V$ be a feasible strong stable set, and let S^* be the minimum cardinality dominating set in G. Then $|SS| \leqslant |S^*|$.*

This duality result can also be viewed as a special case of a known result in the theory of hypergraphs that relates the strong stability number and covering number of a hypergraph [B].

4. The algorithm. Any set S that is both strongly stable in $G(W)$ and dominating in the square of the graph, $G(W)^2$, satisfies inequalities $(*)$. If $W \leqslant w(S^*)$ and S is of size not exceeding k, then, by Lemma 2, we have found our 2-approximate solution. Our algorithm finds a 2-approximate solution by satisfying precisely these constraints.

Before proving the main result, we introduce one final lemma.

LEMMA 4. *Let S be a strong stable set in G. If x is not dominated by S in G^2 then $S \cup \{x\}$ is a strong stable set in G.*

PROOF. Suppose not, i.e. suppose that there is some vertex v such that $U = N_G(v) \cap (S \cup \{x\})$ contains at least two vertices one of which is x. But then x is dominated in G^2 by all other vertices in U. ∎

Algorithm k-center. INPUT: $G = (V, E)$, a complete graph, with $E = \{e_1, e_2, \ldots, e_m\}$, and w_e, for all $e \in E$. (We assume that the edges are ordered such that $w_{e_1} \leqslant w_{e_2} \leqslant \cdots \leqslant w_{e_m}$. Furthermore, we assume that the graph is stored in adjacency list form, where for each vertex, the vertices adjacent to it are listed in increasing edge weight order. Let $G_i = (V, E_i)$ where $E_i = \{e_1, \ldots, e_i\}$.)

OUTPUT: A set S with $|S| \leqslant k$.

begin
if $k = |V|$ **then output** V and **halt**
low := 1 {S can be all V}
high := m {S can be any single $v \in V$}
until *high* = *low* + 1 **do** {binary search}
  begin
    mid := $\lfloor high + low/2 \rfloor$
    {Let ADJ_{mid} denote the adjacency lists for G_{mid}.
    These need not be constructed because given $w_{e_{mid}}$
    and the sorted adjacency lists, we can simulate having them.}
    $S := \emptyset$
    $T := V$
    while $\exists x \in T$ **do**
      begin
        $S := S \cup \{x\}$
        for all $v \in ADJ_{mid}(x)$ **do**
          $T := T - ADJ_{mid}(v) - \{v\}$
      end
    if $|S| \leqslant k$ **then do**
        begin
          high := *mid*
          $S' := S$
        end
      else *low* := *mid*
  end
  output S'
end

THEOREM 1. *Algorithm k-Center produces a set S such that $w(S) \leqslant 2w(S^*)$ in $O(|E| \log |E|)$ time.*

PROOF. Let S be the set produced by the algorithm at the end of a given pass through the **until** loop. We first show that S is both strongly stable in $G_{mid} = (V, E_{mid})$ and dominating in G_{mid}^2. Consider the computation of S for some value of *mid*. At the start of any pass through the **while** loop, S is a strong stable set, and T is the set of vertices not dominated by S. This claim clearly holds initially. Now suppose that the claim is true at the start of the ith pass through the **while** loop; we will show that it is still true at the end of loop. By Lemma 4, it follows that the new S is strongly stable. Furthermore, the vertices dominated by x in G_{mid}^2 are precisely those vertices that we delete from T. Hence it follows that T is still the set of undominated vertices. Therefore, we have shown by induction that the claim holds, and when the algorithm leaves the **while** loop, S must be both a strong stable set in G_{mid} and a dominating set in G_{mid}^2.

We observe that throughout the execution of this algorithm, $w_{e_{low}} < w(S^*)$. This follows from the fact that G_{low} has a strong stable set of size greater than k and thus the minimum dominating set has size larger than k. (Recall that for all $W \geqslant w(S^*)$, $G(W)$ has a dominating set of size at most k.) Consider the values *low* and *high* at termination; at this point $high = low + 1$, so $w_{e_{high}} \leqslant w(S^*)$. The set S' output by the algorithm is a dominating set in G_{high}^2. The performance guarantee of the algorithm follows directly from Lemma 2.

The **until** loop is performing a binary search, and thus is executed at most $\log|E|$ times. To complete the proof we need only note that each edge of the graph is examined at most once (since the other endpoint is deleted from T as soon as the edge is detected) and using straightforward data structures the **while** loop will take $O(|E_{mid}|)$ time. Note that the assumption that the edges are sorted is only a notational convenience, since the time to sort them is of the same order as the remainder of the algorithm. ∎

5. Complexity results. In this section we show that both the dominating set problem for squared graphs and the strong stable set problem are NP-complete.

Dominating set for squared graphs (DS^2). INSTANCE: A graph $G = (V, E)$ and an integer k.

QUESTION: Does the graph G^2 have a dominating set of size k?

THEOREM 2. *DS^2 is NP-complete.*

PROOF. We will reduce from 3-SAT. Suppose that an instance of 3-SAT contains the variables $x_1, x_2, \ldots, x_n$, and has clauses $C_1, \ldots, C_m$. Then construct G as follows. For each variable x_i construct $G_i = (V_i, E_i)$ where $V_i = \{v_{i0}, v_{i1}, \ldots, v_{i5}\}$ and $E_i = \{(v_{i0}, v_{i1}), (v_{i1}, v_{i2}), \ldots, (v_{i4}, v_{i5}), (v_{i5}, v_{i0})\} \cup \{(v_{i3}, v_{i5})\}$. It will be convenient to refer to nodes v_{i0} and v_{i2} as x_i and $\bar{x}_i$, respectively. Let $V_C = \{y_1, \ldots, y_m\}$; that is, each vertex in V_C corresponds to a clause. Let $E_C = \{(y_i, l) \mid l$ is a literal (i.e., x_j or $\bar{x}_j$) in clause $C_i\}$. Let $V = (\bigcup_{i=1}^n V_i) \cup V_C$ and let $E = (\bigcup_{i=1}^n E_i) \cup E_C$. Then $G = (V, E)$ and set $k = n$. It is not hard to verify that G^2 has a dominating set of size k if and only if the original boolean formula was satisfiable. ∎

Next we consider the following problem.

Strong stable set (SSS). INSTANCE: A graph G and an integer k.
QUESTION: Does the graph G contain a strong stable set of size k?

THEOREM 3. *SSS is NP-complete.*

PROOF. We reduce from the ordinary stable set problem. Suppose that the instance of the stable set problem consists of a graph $G' = (V', E')$ and an integer k'. Without loss of generality, we can assume that $k' > 0$ and that G' does not contain any isolated vertices. Let $V' = \{v_1, v_2, \ldots, v_n\}$. If $k' = 1$, let $k = k'$ and $G = G'$. Otherwise, we create G as follows. Start by making k' copies of G', $G_i = (V_i, E_i)$, $i = 1, \ldots, k'$. Let $V_i = \{v_{1i}, v_{2i}, \ldots, v_{ni}\}$. The essential idea of this reduction is that we will insure that every strong stable set in the graph G will have at most one vertex in each copy G_i; every strong stable set will have at most one copy of a given vertex v_i; and finally, if a strong stable set contains v_{ip} and v_{jl} then (v_i, v_j) is not an edge of G'. This is accomplished by adding a number of gadgets to the union of the G_i. For each copy we add a vertex y_i that is adjacent to every vertex in G_i. This insures the first condition. For each $(v_i, v_j) \in E'$ we create a vertex $z_{i,j}$ that is adjacent to v_{il} and v_{jl} for all l. This forces both the second and third conditions. Finally, to insure that the new vertices cannot be used in nontrivial strong stable sets, form a clique on the new vertices. Formally, let $V_C = \{y_1, y_2, \ldots, y_{k'}\} \cup \{z_{i,j} \mid (v_i, v_j) \in E'\}$ and let $V = (\bigcup_{i=1}^{k'} V_i) \cup V_C$. Then set

$$E = \left(\bigcup_{i=1}^{k'} E_i\right) \cup \{(y_l, v_{il}) \mid i = 1, \ldots, n, l = 1, \ldots, k'\}$$

$$\cup \{(z_{i,j}, v_{il}), (z_{i,j}, v_{jl}) \mid (v_i, v_j) \in E', l = 1, \ldots, k'\} \cup \{(w, x) \mid w, x \in V_C\}.$$

It is straightforward to verify that this construction does indeed work. ∎

6. Summary. An interesting aspect of our result is that we identify a set bounded between the dominating set of a graph and a dominating set in the square of the graph. Both of these problems are *NP*-complete; however, our analysis shows how to find such an intermediate set in polynomial time. This approach, and the insight provided by linear programming, might well be used to find approximation algorithms for other *NP*-complete problems.

References

[B] Berge, C. (1973). *Graphs and Hypergraphs*. North-Holland, Amsterdam.

[H1] Hochbaum, D. S. (1984). When Are *NP*-Hard Location Problems Easy? manuscript, November 1982, *Ann. Oper. Res.* **1** 201–214.

[H2] ———. (1982). Easy Solutions for the k-Center Problem or the Dominating Set Problem on Random Graphs. to appear *Ann. Discrete Math.*

[HN] Hsu, W. L. and Nemhauser, G. L. (1979). Easy and Hard Bottleneck Location Problems. *Discrete Appl. Math.* **1** 209–216.

[PR] Parker, R. G. and Rardin, R. L. (1982). On the Bottleneck Travelling Salesman Problem. Industrial and Systems Engineering Report Series J-82-9, Georgia Institute of Technology, October.

SCHOOL OF BUSINESS ADMINISTRATION, UNIVERSITY OF CALIFORNIA, BERKELEY, CALIFORNIA 94720