The k-center problem

Approximation Algorithms 2009
Petros Potikas

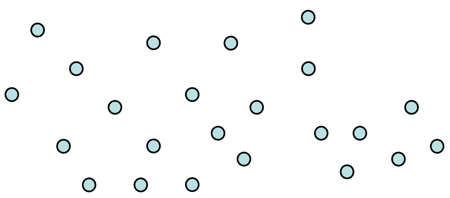
The k-center problem

Definition: Let $G=(V,E)$ be a complete undirected graph with edge costs satisfying the triangle inequality and k be an integer, $0 < k \leq |V|$. For any $S \subseteq V$ and vertex $v \in V$, define *connect(v,S)* to be the cost of the cheapest edge from v to a vertex in S.

Goal: Find a set $S \subseteq V$, with $|S| = k$, so as to

minimize $\max_v\{\text{connect}(v,S)\}$.

Applications: Place k fire stations or warehouses.

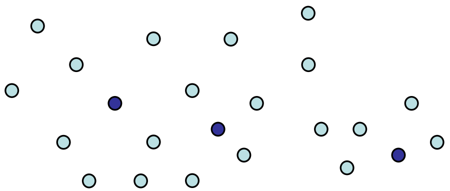


The k-center problem

Definition: Let $G=(V,E)$ be a complete undirected graph with edge costs satisfying the triangle inequality and k be an integer, $0 < k \leq |V|$. For any $S \subseteq V$ and vertex $v \in V$, define *connect(v,S)* to be the cost of the cheapest edge from v to a vertex in S.

Goal: Find a set $S \subseteq V$, with $|S| = k$, so as to minimize $\max_v\{\text{connect}(v,S)\}$.

Applications: Place k fire stations or warehouses.



The k-center problem

Results

- NP-hard problem.
- Approximation algorithm with ratio 2.
- Technique: parametric pruning.
- Generalization to a weighted variant.

The k-center problem

Theorem 1: *It is NP-hard to approximate the general k-center within factor $a(n)$, for any computable function $a(n)$.*

Proof:

Reduction from dominating set… □

The k-center problem

Technique: parametric pruning

Idea: prune irrelevant parts of the input

- Suppose OPT = t
- We want a 2-approximation algorithm
- Any edges of cost more than $2 \cdot t$ are useless: if two vertices are connected by such an edge and one of them gets is picked, then the other vertex is too far away
- We can remove expensive edges

We don’t know OPT, but we guess.

The k-center problem

Technique: parametric pruning

- Order the edges by cost: $\text{cost}(e_1) \leq \text{cost}(e_2) \leq \ldots \leq \text{cost}(e_m)$
- Let $G_i = (V, E_i)$, where $E_i = \{e_1, \ldots, e_i\}$
- The k-center problem is equivalent to finding the minimal index i such that

$$G_i \text{ has a dominating set of size} \leq k$$

- Let i^* be this minimal i
- Then $\text{OPT} = \text{cost}(e_{i^*})$

This is still an NP-hard problem!

The k-center problem

Dominating Set: Let $H=(U,F)$ be an undirected graph. A subset $S \subseteq U$ is a dominating set if every vertex in $U - S$ is adjacent to a vertex in S.

Goal: Find the minimum dominating set in H.

Dominating Set is NP-hard.

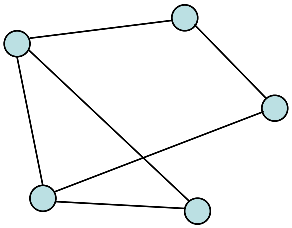
Independent Set: Let $H=(U,F)$ be an undirected graph. A subset $S \subseteq U$ is an independent set if there is no edge in H having both ends in S.

Maximum Independent Set is NP-hard.

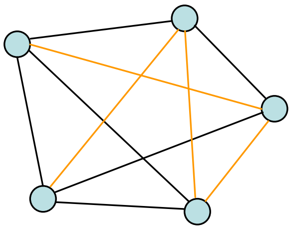
The k-center problem

Powers of graphs

Let G=(V,E) be a graph. The square of G is the graph G^2=(V,E'), where (u,v) $\in$ E' if there is a path of length at most 2 between u and v in G (and $u \neq v$).



G



G^2

Generalization: G^t

The k-center problem

Lemma 1: *Given a graph G, let I be an independent set in* G^2.
Then, $|I| \leq \mathrm{dom}(G)$.

Proof:
Let D be a minimum dominating set in G $(|D|=dom(G))$.
Then G contains $|D|$ stars spanning all vertices of G (the vertices of D are the centers of the stars).
A star in G becomes a clique in G^2.
So G^2 contains $|D|$ cliques spanning all vertices.
Independent set I can pick at most one vertex from each clique. □

The k-center problem

Algorithm 1 (Metric k-center)

We use that maximal independent sets can be found in polynomial time.

1. Construct $G_1^2, G_2^2, \ldots, G_m^2$.
2. Compute a maximal independent set, M_i, in each graph G_i^2.
3. Find the smallest index i, such that $|M_i| \leq k$, say j.
4. Return M_j.

The k-center problem

Lemma 2: *For j as defined in the above algorithm*, $\text{cost}(e_j) \leq \text{OPT}$.

Proof:

- For $i < j$, we have that $|M_i| > k$.
- By Lemma 1, $\text{dom}(G_i) > k$.
- So, $i^* > i$.

Thus, $j \leq i^*$. □

The k-center problem

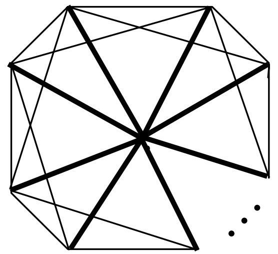
Theorem 2: *Algorithm metric k-center achieves an approximation factor* 2.

Proof:

- Any maximal independent set I in G_j^2 is also a dominating set (for, if some vertex u is not dominated by I, then $I \cup \{u\}$ is an independent set, contradicting I's maximality).
- In G_j^2 we have $|M_j|$ stars centered on the vertices in M_j.
- These stars cover all the vertices.
- Each edge used in constructing these stars has cost at most $2 \cdot \text{cost}(e_j) \leq 2 \cdot \text{OPT}$ (by Lemma 2). □

The k-center problem

Tight example:



n+1 vertices

thick edges have cost 1, all edges incident to the center

thin edges have cost 2, the rest of the edges (not all edges of cost 2 are shown)

For k = 1, OPT = 1, the center of the wheel

The algorithm will compute j=n, $G_n^{\,2}$ is a clique, and if a peripheral vertex is chosen, then cost is 2.

The k-center problem

Theorem 3: *If* P ≠ NP, *no approximation algorithm gives a* (2-ε)-*approximation for* ε>0.

Proof:

Reduction from dominating set to the metric k-center problem.

Let $G = (V,E)$, k be an instance of the dominating set problem.

We define the complete graph $G' = (V,E')$, where

$$\text{cost}(u,v)=1, \text{ if } (u,v) \in E$$
$$\text{cost}(u,v)=2, \text{ if } (u,v) \notin E$$

G' satisfies the triangle inequality.

The k-center problem

Theorem 3: *If* P $\neq$ NP, *no approximation algorithm gives a* (2-ε)-*approximation for* ε>0.

Proof(cont'd):

Suppose *G* has a dominating set of size at most *k*.

Then *G'* has a k-center of cost 1

→a (2-ε)-approximation algorithm delivers one with cost < 2.

If there is no such dominating set in *G*, every k-center has cost ≥ 2 > 2-ε.

Thus, a (2-ε)-approximation algorithm for the k-center problem can be used to determine whether or not there is a dominating set of size *k*. □

The weighted k-center problem

***Definition*:** Let $G=(V,E)$ be a complete undirected graph with edge costs satisfying the triangle inequality, with weights on vertices and a bound $W \in R^+$. For any $S \subseteq V$ and vertex $v \in V$, define *connect(v,S)* to be the cost of the cheapest edge from v to a vertex in S.

***Goal*:** Find a set $S \subseteq V$, with total weight at most W, so as to minimize $\max_v\{\text{connect}(v,S)\}$.

Applications: Place fire stations or warehouses, given a budget.

The weighted k-center problem

- We use the same graphs $G_1, G_2, \ldots, G_m$
- Let wdom(G) be the weight of a minimum weight dominating set in G
- Find the minimal index i such that

$$\mathrm{wdom}(G_i) \leq W$$

- Let i^* be this minimal i
- Then OPT=cost(e_{i^*})

The weighted k-center problem

- Let I be an independent set in G^2
- For any vertex u, let $s(u)$ denote its lightest neighbor of u
- We also consider u to be a neighbor of itself
- Let $S=\{s(u) \mid u \in I\}$
- We claim $w(S) \leq \mathrm{wdom}(G)$

The weighted k-center problem

Lemma 3: $w(S) \leq \text{wdom}(G)$

Proof:

Let D be a minimum weight dominating set in G ($w(D)$=wdom(G)).

Then G contains $|D|$ stars spanning all vertices of G (the vertices of D are the centers of the stars).

A star in G becomes a clique in G^2.

So G^2 contains $|D|$ cliques spanning all vertices.

Independent set I can pick at most one vertex from each clique.

Each vertex in I has the center of the corresponding star available as a neighbor in G (this might not be the lightest neighbor).

Thus, $w(S) \leq \text{wdom}(G)$. □

The weighted k-center problem

Algorithm 2 (Weighted k-center)

Let $s_i(u)$ denote the lightest neighbor of u in G_i.

1. Construct $G_1{}^2, G_2{}^2, \ldots, G_m{}^2$
2. Compute a maximal independent set, M_i, in each graph G_i^2
3. Compute $S_i = \{s_i(u) \mid u \in M_i\}$
4. Find the smallest index i, such that $w(S_i) \leq W$, say j
5. Return S_j

The weighted k-center problem

Theorem 2: *This algorithm achieves a 3-approximation.*

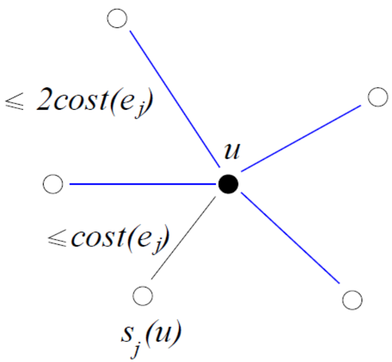
Proof:

OPT $\geq$ cost(e_j) (as Lemma 2)

M_j is a dominating set in G_j^2

We can cover V with stars of G_j^2 centered in vertices of M_j

These stars use edges of cost at most 2·cost(e_j)



The weighted k-center problem

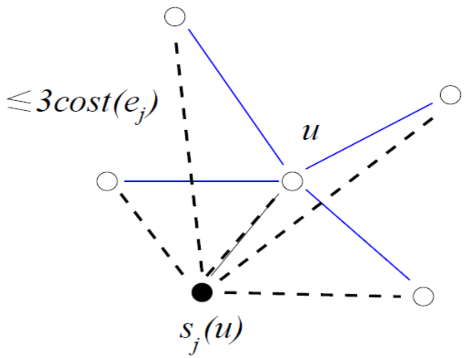
Theorem 2: *This algorithm achieves a 3-approximation.*

Proof (cont'd):

Each star center is adjacent to a vertex in S_j, using an edge of cost at most cost(e_j)

Move each center of these stars to the adjacent vertex in S_j and redefine the star

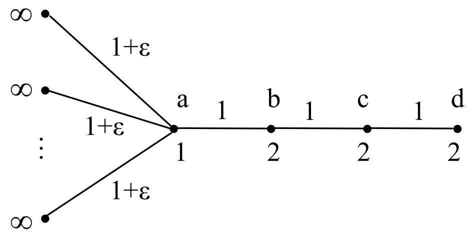
Every vertex in G_j, can be reached by a cost at most 3·cost(e_j)



□

The weighted k-center problem

Tight example:



n+4 vertices, W=3

All edges not shown have cost equal to the cost of the shortest path in the graph shown.

OPT = 1+ε ($\{a,c\}$)

For any $i<n+3$, set G_i is missing at least one edge of cost 1+ε.

One vertex will be isolated (also in G_i^2) so it will be in S_i.

For $i=n+3$, $\{b\}$ is a maximal independent set. If the algorithm chooses $\{b\}$, then the center of the star will be $S_{n+3}=\{a\}$, with cost=3.

The k-center problem

Related problem

Metric k-cluster: Let $G=(V,E)$ be a complete undirected graph with edge costs satisfying the triangle inequality and k be an integer, $0 < k \leq |V|$.

Goal: Partition V into sets $V_1,V_2,\ldots,V_k$, so as to minimize the costliest edge between two vertices in the same set, i.e. minimize

$$max_{1\leq i\leq k\ u,v\in Vi}\ cost(u,v)$$