The center selection problem

- We are given a set S of n sites, an integer k, and also distances $d(s, z)$ between every two sites s and z.

The center selection problem

- We are given a set S of n sites, an integer k, and also distances $d(s, z)$ between every two sites s and z.
- The distances satisfy:
  - $\forall s \in S,\ d(s, s) = 0$;
  - $\forall s, z \in S,\ d(s, z) = d(z, s)$.
  - $\forall s, z, h \in S,\ d(s, z) + d(z, h) \geq d(s, h)$.

The center selection problem

- We are given a set S of n sites, an integer k, and also distances $d(s, z)$ between every two sites s and z.
- The distances satisfy:
  - $\forall s \in S,\ d(s, s) = 0$;
  - $\forall s, z \in S,\ d(s, z) = d(z, s)$.
  - $\forall s, z, h \in S,\ d(s, z) + d(z, h) \geq d(s, h)$.
- For a set $C \subseteq S$ of *centers*, the distance from a site s to C is $d(s, C) := \min_{c \in C} d(s, c)$.

The center selection problem

- We are given a set S of n sites, an integer k, and also distances $d(s, z)$ between every two sites s and z.
- The distances satisfy:
  - $\forall s \in S,\ d(s, s) = 0$;
  - $\forall s, z \in S,\ d(s, z) = d(z, s)$.
  - $\forall s, z, h \in S,\ d(s, z) + d(z, h) \geq d(s, h)$.
- For a set $C \subseteq S$ of *centers*, the distance from a site s to C is $d(s, C) := \min_{c \in C} d(s, c)$.
- The covering radius of C is $\max_{s \in S} d(s, C)$.

The center selection problem

- We are given a set S of n sites, an integer k, and also distances $d(s, z)$ between every two sites s and z.
- The distances satisfy:
  - $\forall s \in S,\ d(s, s) = 0$;
  - $\forall s, z \in S,\ d(s, z) = d(z, s)$.
  - $\forall s, z, h \in S,\ d(s, z) + d(z, h) \geq d(s, h)$.
- For a set $C \subseteq S$ of *centers*, the distance from a site s to C is $d(s, C) := \min_{c \in C} d(s, c)$.
- The covering radius of C is $\max_{s \in S} d(s, C)$.
- We are asked to choose a set of k centers to minimize its covering radius.

The center selection problem

- We are given a set S of n sites, an integer k, and also distances $d(s, z)$ between every two sites s and z.
- The distances satisfy:
  - $\forall s \in S,\ d(s, s) = 0$;
  - $\forall s, z \in S,\ d(s, z) = d(z, s)$.
  - $\forall s, z, h \in S,\ d(s, z) + d(z, h) \geq d(s, h)$.
- For a set $C \subseteq S$ of *centers*, the distance from a site s to C is $d(s, C) := \min_{c \in C} d(s, c)$.
- The covering radius of C is $\max_{s \in S} d(s, C)$.
- We are asked to choose a set of k centers to minimize its covering radius.
- The problem is NP-hard. (Reduction?)

A Greedy Algorithm

- A natural, intuitive greedy approach: find a site that is "central" to many sites, set up a center there, remove all sites close to it (i.e., consider them as covered), and repeat the procedure.

A Greedy Algorithm

- A natural, intuitive greedy approach: find a site that is "central" to many sites, set up a center there, remove all sites close to it (i.e., consider them as covered), and repeat the procedure.
- Problem with the approach: How do we compare "centrality" of two sites? And which sites should be considered "covered"?

A Greedy Algorithm

- A natural, intuitive greedy approach: find a site that is "central" to many sites, set up a center there, remove all sites close to it (i.e., consider them as covered), and repeat the procedure.
- Problem with the approach: How do we compare "centrality" of two sites? And which sites should be considered "covered"?
- Both problems would be solved if we have a covering radius to shoot for: suppose we are interested in whether it is possible to choose k centers with covering radius $\leq r$:

A Greedy Algorithm

- A natural, intuitive greedy approach: find a site that is "central" to many sites, set up a center there, remove all sites close to it (i.e., consider them as covered), and repeat the procedure.
- Problem with the approach: How do we compare "centrality" of two sites? And which sites should be considered "covered"?
- Both problems would be solved if we have a covering radius to shoot for: suppose we are interested in whether it is possible to choose k centers with covering radius $\leq r$:
- The greedy procedure:
  - Itialize $R \leftarrow S$, $C \leftarrow \emptyset$. (R will be the set of sites "not covered" yet, and C will be the set of centers we choose.)

A Greedy Algorithm

- A natural, intuitive greedy approach: find a site that is "central" to many sites, set up a center there, remove all sites close to it (i.e., consider them as covered), and repeat the procedure.
- Problem with the approach: How do we compare "centrality" of two sites? And which sites should be considered "covered"?
- Both problems would be solved if we have a covering radius to shoot for: suppose we are interested in whether it is possible to choose k centers with covering radius $\leq r$:
- The greedy procedure:
  - Itialize $R \leftarrow S$, $C \leftarrow \emptyset$. (R will be the set of sites "not covered" yet, and C will be the set of centers we choose.)
  - While R is nonempty and $|C| < k$, do: add an arbitrary site $s \in R$ to C, remove from R any site within distance r to s.

A Greedy Algorithm

- A natural, intuitive greedy approach: find a site that is "central" to many sites, set up a center there, remove all sites close to it (i.e., consider them as covered), and repeat the procedure.
- Problem with the approach: How do we compare "centrality" of two sites? And which sites should be considered "covered"?
- Both problems would be solved if we have a covering radius to shoot for: suppose we are interested in whether it is possible to choose k centers with covering radius $\leq r$:
- The greedy procedure:
  - Itialize $R \leftarrow S$, $C \leftarrow \emptyset$. (R will be the set of sites "not covered" yet, and C will be the set of centers we choose.)
  - While R is nonempty and $|C| < k$, do: add an arbitrary site $s \in R$ to C, remove from R any site within distance r to s.
  - If we terminate with a non-empty R, declare failure; otherwise we find a set C, $|C| \leq k$, with a covering radius $\leq r$.

A Greedy Algorithm

- A natural, intuitive greedy approach: find a site that is "central" to many sites, set up a center there, remove all sites close to it (i.e., consider them as covered), and repeat the procedure.
- Problem with the approach: How do we compare "centrality" of two sites? And which sites should be considered "covered"?
- Both problems would be solved if we have a covering radius to shoot for: suppose we are interested in whether it is possible to choose k centers with covering radius $\leq r$:
- The greedy procedure:
  - Itialize $R \leftarrow S$, $C \leftarrow \emptyset$. (R will be the set of sites "not covered" yet, and C will be the set of centers we choose.)
  - While R is nonempty and $|C| < k$, do: add an arbitrary site $s \in R$ to C, remove from R any site within distance r to s.
  - If we terminate with a non-empty R, declare failure; otherwise we find a set C, $|C| \leq k$, with a covering radius $\leq r$.
- Note that the algorithm isn't fully "greedy": in each step s is chosen arbitrarily.

Analysis

The terminating condition does *not* say that, if the algorithm fails, there is no C, $|C| \leq k$, with covering radius $\leq r$. (Otherwise we can just try all r's and have a polynomial-time algorithm for the problem.)

Analysis

The terminating condition does *not* say that, if the algorithm fails, there is no C, $|C| \leq k$, with covering radius $\leq r$. (Otherwise we can just try all r's and have a polynomial-time algorithm for the problem.)

Theorem

If the above greedy algorithm fails, for any $C^* \subseteq S$ with $|C^*| \leq k$, the covering radius of C^* is at least $r/2$.

Analysis

The terminating condition does *not* say that, if the algorithm fails, there is no C, $|C| \leq k$, with covering radius $\leq r$. (Otherwise we can just try all r's and have a polynomial-time algorithm for the problem.)

Theorem

If the above greedy algorithm fails, for any $C^* \subseteq S$ with $|C^*| \leq k$, the covering radius of C^* is at least $r/2$.

Proof.

Let C^* be any subset of S with covering radius $< \frac{r}{2}$, we show $|C^*| > k$.

Analysis

The terminating condition does *not* say that, if the algorithm fails, there is no C, $|C| \leq k$, with covering radius $\leq r$. (Otherwise we can just try all r's and have a polynomial-time algorithm for the problem.)

Theorem

If the above greedy algorithm fails, for any $C^* \subseteq S$ with $|C^*| \leq k$, the covering radius of C^* is at least $r/2$.

Proof.

Let C^* be any subset of S with covering radius $< \frac{r}{2}$, we show $|C^*| > k$. Recall our algorithm terminated with a set of centers C, $|C| = k$, without covering all sites within distance r.

Proof (continued..)

For any site s and radius δ, let's denote by $B(s, \delta)$ the set of sites within distance δ to s, i.e., $B(s, \delta) \coloneqq \{t \in S : d(s, t) \leq \delta\}$.

Proof (continued..)

For any site s and radius δ, let's denote by $B(s, \delta)$ the set of sites within distance δ to s, i.e., $B(s, \delta) \coloneqq \{t \in S : d(s, t) \leq \delta\}$.
Consider any $c \in C$. There must be some $o_c \in C^*$ that is in $B(c, \frac{r}{2})$.

Proof (continued..)

For any site s and radius δ, let's denote by $B(s, \delta)$ the set of sites within distance δ to s, i.e., $B(s, \delta) \coloneqq \{t \in S : d(s, t) \leq \delta\}$.
Consider any $c \in C$. There must be some $o_c \in C^*$ that is in $B(c, \frac{r}{2})$.
Key observation: $B(o_c, \frac{r}{2}) \subseteq B(c, r)$.

Proof (continued..)

For any site s and radius δ, let's denote by $B(s, \delta)$ the set of sites within distance δ to s, i.e., $B(s, \delta) \coloneqq \{t \in S : d(s, t) \leq \delta\}$.
Consider any $c \in C$. There must be some $o_c \in C^*$ that is in $B(c, \frac{r}{2})$.
Key observation: $B(o_c, \frac{r}{2}) \subseteq B(c, r)$.
Because $\forall s \in B(o_c, \frac{r}{2})$,

$$d(s, c) \leq d(s, o_c) + d(o_c, c) \leq \frac{r}{2} + \frac{r}{2} = r.$$

Proof (continued..)

For any site s and radius δ, let's denote by $B(s, \delta)$ the set of sites within distance δ to s, i.e., $B(s, \delta) := \{t \in S : d(s, t) \leq \delta\}$.
Consider any $c \in C$. There must be some $o_c \in C^*$ that is in $B(c, \frac{r}{2})$.
Key observation: $B(o_c, \frac{r}{2}) \subseteq B(c, r)$.
Because $\forall s \in B(o_c, \frac{r}{2})$,

$$d(s, c) \leq d(s, o_c) + d(o_c, c) \leq \frac{r}{2} + \frac{r}{2} = r.$$

Therefore, for any $c, c' \in C$, $o_c \neq o_{c'}$.

Proof (continued..)

For any site s and radius δ, let's denote by $B(s, \delta)$ the set of sites within distance δ to s, i.e., $B(s, \delta) \coloneqq \{t \in S : d(s, t) \leq \delta\}$.
Consider any $c \in C$. There must be some $o_c \in C^*$ that is in $B(c, \frac{r}{2})$.
Key observation: $B(o_c, \frac{r}{2}) \subseteq B(c, r)$.
Because $\forall s \in B(o_c, \frac{r}{2})$,

$$d(s, c) \leq d(s, o_c) + d(o_c, c) \leq \frac{r}{2} + \frac{r}{2} = r.$$

Therefore, for any $c, c' \in C$, $o_c \neq o_{c'}$.
Also, $\cup_{c \in C} B(o_c, \frac{r}{2}) \leq \cup_{c \in C} B(c, r) \subsetneq S$;

Proof (continued..)

For any site s and radius δ, let's denote by $B(s, \delta)$ the set of sites within distance δ to s, i.e., $B(s, \delta) \coloneqq \{t \in S : d(s, t) \le \delta\}$.
Consider any $c \in C$. There must be some $o_c \in C^*$ that is in $B(c, \frac{r}{2})$.
Key observation: $B(o_c, \frac{r}{2}) \subseteq B(c, r)$.
Because $\forall s \in B(o_c, \frac{r}{2})$,

$$d(s, c) \le d(s, o_c) + d(o_c, c) \le \frac{r}{2} + \frac{r}{2} = r.$$

Therefore, for any $c, c' \in C$, $o_c \ne o_{c'}$.
Also, $\cup_{c \in C} B(o_c, \frac{r}{2}) \le \cup_{c \in C} B(c, r) \subsetneq S$;
Now since $\cup_{o \in C^*} B(o, \frac{r}{2}) = S$ by assumption, $|C^*| > |C| = k$. □

The final algorithm

- Getting rid of the dependance on r: a binary search would do.
  - If the optimal solution has covering radius r^*, the algorithm must succeed when it tries any radius $r \geq 2r^*$.

The final algorithm

- Getting rid of the dependance on r: a binary search would do.
  - If the optimal solution has covering radius r^*, the algorithm must succeed when it tries any radius $r \geq 2r^*$.
  - This gives rise to a 2-approximation algorithm.

The final algorithm

- Getting rid of the dependance on r: a binary search would do.
  - If the optimal solution has covering radius r^*, the algorithm must succeed when it tries any radius $r \geq 2r^*$.
  - This gives rise to a 2-approximation algorithm.
- For this problem, we can do something a bit more clever:

The final algorithm

- Getting rid of the dependance on r: a binary search would do.
  - If the optimal solution has covering radius r^*, the algorithm must succeed when it tries any radius $r \geq 2r^*$.
  - This gives rise to a 2-approximation algorithm.
- For this problem, we can do something a bit more clever:
  - Pick an arbitrary site s and itinialize $C \leftarrow \{s\}$.

The final algorithm

- Getting rid of the dependance on r: a binary search would do.
  - If the optimal solution has covering radius r^*, the algorithm must succeed when it tries any radius $r \geq 2r^*$.
  - This gives rise to a 2-approximation algorithm.
- For this problem, we can do something a bit more clever:
  - Pick an arbitrary site s and itinialize $C \leftarrow \{s\}$.
  - While $|C| < k$, do: pick a site c that maximizes $d(c, C)$, add c to C.

The final algorithm

- Getting rid of the dependance on r: a binary search would do.
  - If the optimal solution has covering radius r^*, the algorithm must succeed when it tries any radius $r \geq 2r^*$.
  - This gives rise to a 2-approximation algorithm.
- For this problem, we can do something a bit more clever:
  - Pick an arbitrary site s and itinialize $C \leftarrow \{s\}$.
  - While $|C| < k$, do: pick a site c that maximizes $d(c, C)$, add c to C.

Claim

This new algorithm gives a 2-approximation to the minimum covering radius.

The final algorithm

- Getting rid of the dependance on r: a binary search would do.
  - If the optimal solution has covering radius r^*, the algorithm must succeed when it tries any radius $r \geq 2r^*$.
  - This gives rise to a 2-approximation algorithm.
- For this problem, we can do something a bit more clever:
  - Pick an arbitrary site s and itinialize $C \leftarrow \{s\}$.
  - While $|C| < k$, do: pick a site c that maximizes $d(c, C)$, add c to C.

Claim

This new algorithm gives a 2-approximation to the minimum covering radius.

Reason: Let the optimal covering radius be r^*, then the covering radius of C can't be more than $2r^*$.

The final algorithm

- Getting rid of the dependance on r: a binary search would do.
  - If the optimal solution has covering radius r^*, the algorithm must succeed when it tries any radius $r \geq 2r^*$.
  - This gives rise to a 2-approximation algorithm.
- For this problem, we can do something a bit more clever:
  - Pick an arbitrary site s and itinialize $C \leftarrow \{s\}$.
  - While $|C| < k$, do: pick a site c that maximizes $d(c, C)$, add c to C.

Claim

This new algorithm gives a 2-approximation to the minimum covering radius.

Reason: Let the optimal covering radius be r^*, then the covering radius of C can't be more than $2r^*$.
Otherwise the algorithm is an implementation of the previous algorithm with a radius $r > 2r^*$, and yet does not cover all sites within distance r, contradicting the theorem.

Discussion

Can we find an algorithm with better approximation ratio?

Discussion

Can we find an algorithm with better approximation ratio?
Answer: It's NP-hard to get $2-\epsilon$-approximation for any $\epsilon > 0$. (Think about the reduction from dominating set.)