A Randomized O(log n)-Competitive Algorithm for the Online Connected Facility Location Problem

Mário César San Felice[1]
Orlando Lee
David P. Williamson

UNICAMP
Institute of Computing
Laboratory of Optimization and Combinatorics

October 31th, 2014

[1] grant No. 2009/15535-1, São Paulo Research Foundation (FAPESP)

Preliminaries

First we show some problems that are useful to understand the Online Connected Facility Location problem.

Starting with the Steiner Tree problem.

Preliminaries

First we show some problems that are useful to understand the Online Connected Facility Location problem.

Starting with the Steiner Tree problem.

Steiner Tree Problem

Minimization problem whose input is a graph with costs on the edges, a set of terminal nodes and a set of Steiner nodes.

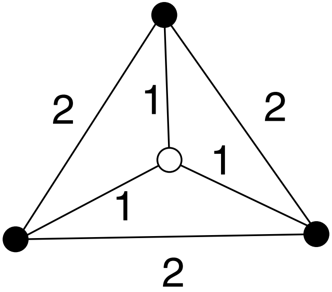
A feasible solution is a tree that connects all terminal nodes and its cost is the sum of the edge costs in the tree.

Steiner Tree Problem

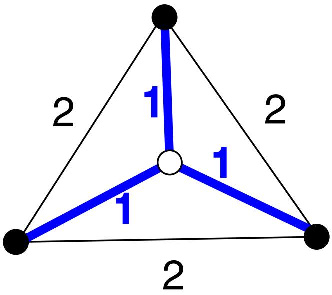
Minimization problem whose input is a graph with costs on the edges, a set of terminal nodes and a set of Steiner nodes.

A feasible solution is a tree that connects all terminal nodes and its cost is the sum of the edge costs in the tree.

Steiner Tree Problem (ex.)



Steiner Tree Problem (ex.)



Rent-or-Buy Problems

Problems in which there is some resource that the algorithm can rent or buy.

A rented resource can be used only once.

A bought resource can be used several times, but its cost is greater than the renting cost.

As an example, lets take the Single Source Rent-or-Buy problem.

Rent-or-Buy Problems

Problems in which there is some resource that the algorithm can rent or buy.

A rented resource can be used only once.

A bought resource can be used several times, but its cost is greater than the renting cost.

As an example, lets take the Single Source Rent-or-Buy problem.

Rent-or-Buy Problems

Problems in which there is some resource that the algorithm can rent or buy.

A rented resource can be used only once.

A bought resource can be used several times, but its cost is greater than the renting cost.

As an example, lets take the Single Source Rent-or-Buy problem.

Rent-or-Buy Problems

Problems in which there is some resource that the algorithm can rent or buy.

A rented resource can be used only once.

A bought resource can be used several times, but its cost is greater than the renting cost.

As an example, lets take the Single Source Rent-or-Buy problem.

Single Source Rent-or-Buy Problem

A rent-or-buy version of the Steiner Tree problem in which all terminals must be connected to a source r.

The algorithm can decide between renting or buying edges.

A rented edge can only be used by one terminal.

A bought edge can be used by all terminals, but its cost is multiplied by M.

Single Source Rent-or-Buy Problem

A rent-or-buy version of the Steiner Tree problem in which all terminals must be connected to a source r.

The algorithm can decide between renting or buying edges.

A rented edge can only be used by one terminal.

A bought edge can be used by all terminals, but its cost is multiplied by M.

Single Source Rent-or-Buy Problem

A rent-or-buy version of the Steiner Tree problem in which all terminals must be connected to a source r.

The algorithm can decide between renting or buying edges.

A rented edge can only be used by one terminal.

A bought edge can be used by all terminals, but its cost is multiplied by M.

Single Source Rent-or-Buy Problem

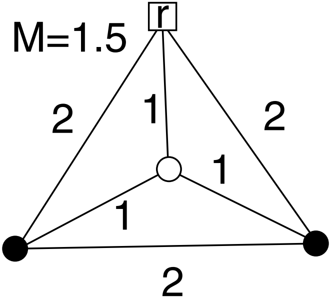
A rent-or-buy version of the Steiner Tree problem in which all terminals must be connected to a source r.

The algorithm can decide between renting or buying edges.

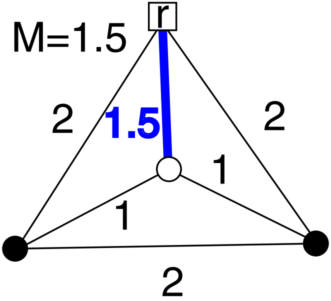
A rented edge can only be used by one terminal.

A bought edge can be used by all terminals, but its cost is multiplied by M.

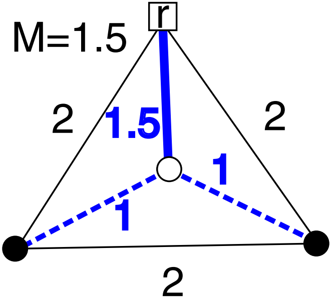
Single Source Rent-or-Buy Problem (ex.)



Single Source Rent-or-Buy Problem (ex.)



Single Source Rent-or-Buy Problem (ex.)



Online Problems

Problems in which the parts of the input arrive one at a time and each part need to be served before the next one arrives.

Also, no decision made to serve a part may be changed in the future.

Online Problems

Problems in which the parts of the input arrive one at a time and each part need to be served before the next one arrives.

Also, no decision made to serve a part may be changed in the future.

Competitive Analysis

Worst case technique used to analyse online algorithms.

We say that an online algorithm ALG is c-competitive if, for every input I and some α constant, we have that:

$$\mathrm{ALG}(I) \leq c\mathrm{OPT}(I) + \alpha.$$

Competitive Analysis

Worst case technique used to analyse online algorithms.

We say that an online algorithm ALG is c-competitive if, for every input I and some α constant, we have that:

$$\mathrm{ALG}(I) \leq c\mathrm{OPT}(I) + \alpha.$$

Online Steiner Tree Problem

This problem is defined similarly to the Steiner Tree problem, except that the terminal nodes arrive one at a time.

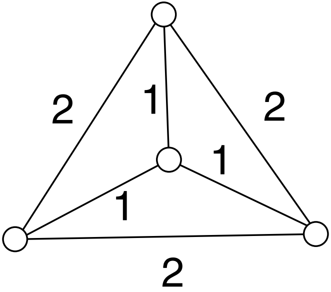
Also, at all times the terminals must be connected by a tree and no edge used may be removed in the future.

Online Steiner Tree Problem

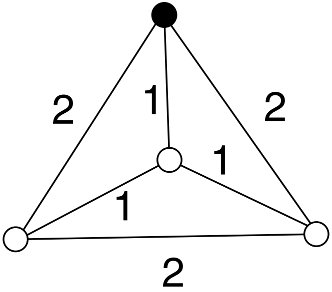
This problem is defined similarly to the Steiner Tree problem, except that the terminal nodes arrive one at a time.

Also, at all times the terminals must be connected by a tree and no edge used may be removed in the future.

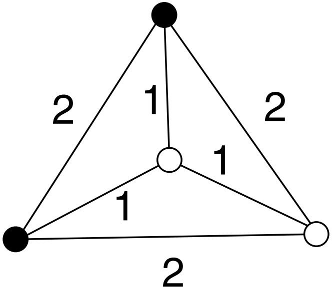
Online Steiner Tree Problem (ex.)



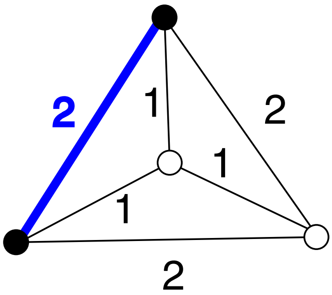
Online Steiner Tree Problem (ex.)



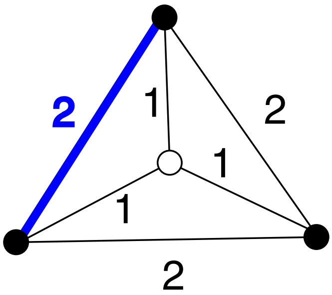
Online Steiner Tree Problem (ex.)



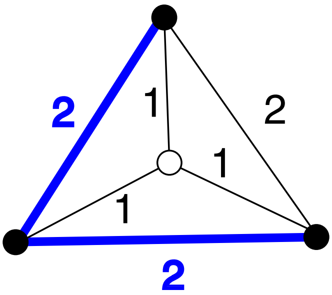
Online Steiner Tree Problem (ex.)



Online Steiner Tree Problem (ex.)



Online Steiner Tree Problem (ex.)



Online Steiner Tree Problem (cont.)

There are $\mathrm{O}(\log n)$-competitive algorithms for the Online Steiner Tree problem.

In particular, there is an algorithm due to Imase and Waxman that has this competitivity and is used in our result.

Also, it is known a $\Omega(\log n)$ lower bound to the competitive ratio of any algorithm for this problem.

Online Steiner Tree Problem (cont.)

There are $\mathrm{O}(\log n)$-competitive algorithms for the Online Steiner Tree problem.

In particular, there is an algorithm due to Imase and Waxman that has this competitivity and is used in our result.

Also, it is known a $\Omega(\log n)$ lower bound to the competitive ratio of any algorithm for this problem.

Online Steiner Tree Problem (cont.)

There are $\mathrm{O}(\log n)$-competitive algorithms for the Online Steiner Tree problem.

In particular, there is an algorithm due to Imase and Waxman that has this competitivity and is used in our result.

Also, it is known a $\Omega(\log n)$ lower bound to the competitive ratio of any algorithm for this problem.

The Facility Location Problem

In this problem the algorithm have to serve clients in a metric space by connecting them to facilities.

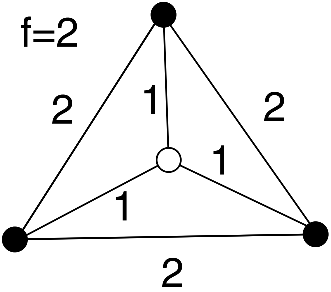
The goal is to minimize the sum of the distances between clients and facilities (connection cost) plus the sum of the facilities costs (opening cost).

The Facility Location Problem

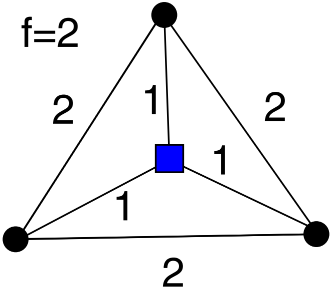
In this problem the algorithm have to serve clients in a metric space by connecting them to facilities.

The goal is to minimize the sum of the distances between clients and facilities (connection cost) plus the sum of the facilities costs (opening cost).

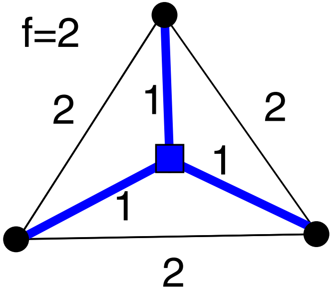
The Facility Location Problem (ex.)



The Facility Location Problem (ex.)



The Facility Location Problem (ex.)



The Online Facility Location Problem

In the online version of the Facility Location problem the clients arrive one at a time.

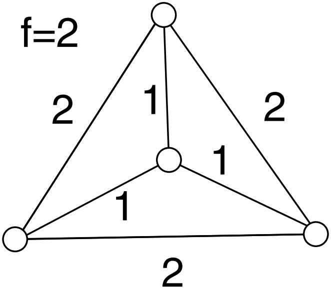
Also, no opened facility can be closed in the future, nor the connection between a client and a facility can be changed.

The Online Facility Location Problem

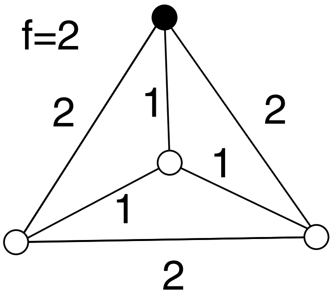
In the online version of the Facility Location problem the clients arrive one at a time.

Also, no opened facility can be closed in the future, nor the connection between a client and a facility can be changed.

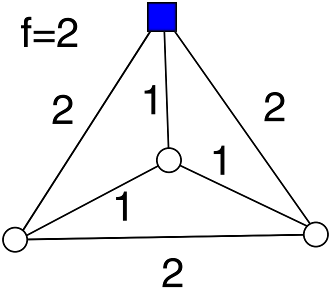
The Online Facility Location Problem (ex.)



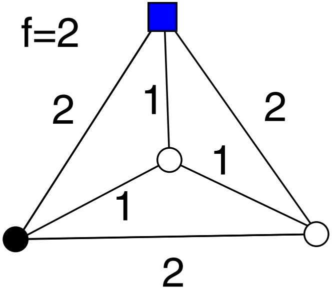
The Online Facility Location Problem (ex.)



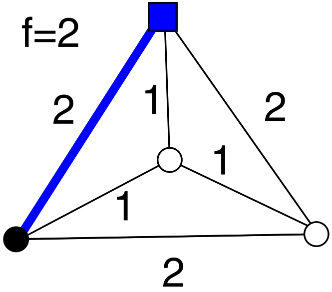
The Online Facility Location Problem (ex.)



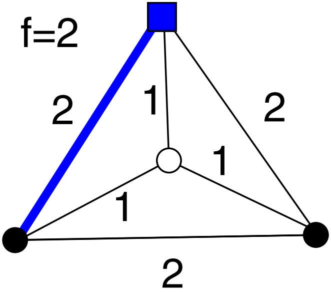
The Online Facility Location Problem (ex.)



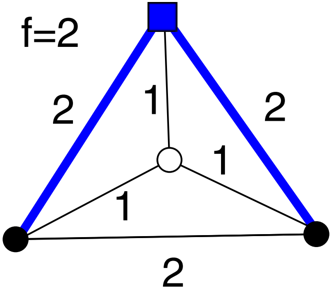
The Online Facility Location Problem (ex.)



The Online Facility Location Problem (ex.)



The Online Facility Location Problem (ex.)



The Online Facility Location Problem (cont.)

There are $\mathrm{O}(\log n)$-competitive algorithms known for this problem.

In particular, there is a primal-dual algorithm due to Fotakis that has this competitivity and is used in our result.

Also, it is known a $\Omega(\frac{\log n}{\log \log n})$ lower bound to the competitive ratio of any algorithm for this problem.

The Online Facility Location Problem (cont.)

There are $O(\log n)$-competitive algorithms known for this problem.

In particular, there is a primal-dual algorithm due to Fotakis that has this competitivity and is used in our result.

Also, it is known a $\Omega(\frac{\log n}{\log \log n})$ lower bound to the competitive ratio of any algorithm for this problem.

The Online Facility Location Problem (cont.)

There are $\mathrm{O}(\log n)$-competitive algorithms known for this problem.

In particular, there is a primal-dual algorithm due to Fotakis that has this competitivity and is used in our result.

Also, it is known a $\Omega(\frac{\log n}{\log \log n})$ lower bound to the competitive ratio of any algorithm for this problem.

Connected Facility Location Problem

This problem is a combination of the Facility Location problem with the Steiner Tree problem.

There is a set of clients that needs to be connected to facilities. Also, the opened facilities need to be connected to each other by a tree T. Each edge of T costs M times the regular cost of it.

The goal is to minimize the total cost of connecting clients, opening facilities and building the tree.

$$\sum_{j \in D} d(j, F^a) + \sum_{i \in F^a} f(i) + M \sum_{e \in T} d(e)$$

Connected Facility Location Problem

This problem is a combination of the Facility Location problem with the Steiner Tree problem.

There is a set of clients that needs to be connected to facilities. Also, the opened facilities need to be connected to each other by a tree T. Each edge of T costs M times the regular cost of it.

The goal is to minimize the total cost of connecting clients, opening facilities and building the tree.

$$\sum_{j \in D} d(j, F^a) + \sum_{i \in F^a} f(i) + M \sum_{e \in T} d(e)$$

Connected Facility Location Problem

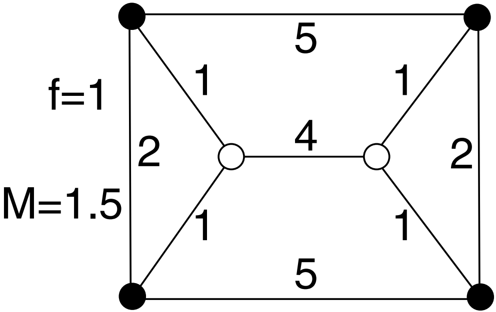
This problem is a combination of the Facility Location problem with the Steiner Tree problem.

There is a set of clients that needs to be connected to facilities. Also, the opened facilities need to be connected to each other by a tree T. Each edge of T costs M times the regular cost of it.

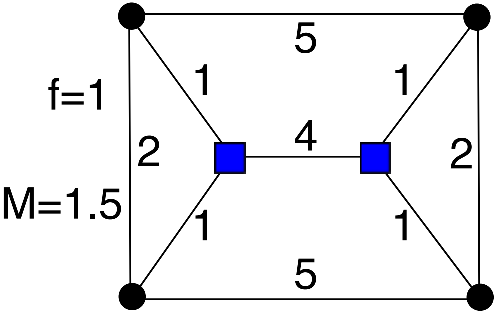
The goal is to minimize the total cost of connecting clients, opening facilities and building the tree.

$$\sum_{j \in D} d(j, F^a) + \sum_{i \in F^a} f(i) + M \sum_{e \in T} d(e)$$

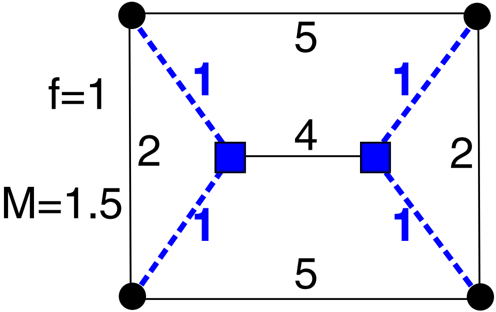
Connected Facility Location Problem (ex.)



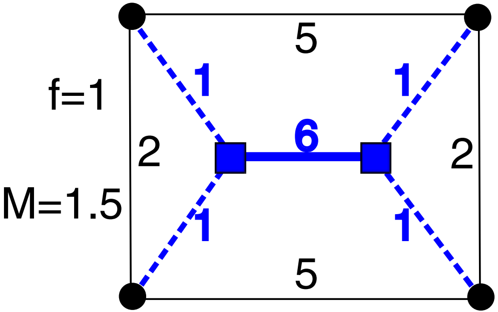
Connected Facility Location Problem (ex.)



Connected Facility Location Problem (ex.)



Connected Facility Location Problem (ex.)



Online Connected Facility Location Problem

In the online version of the Connected Facility Location problem the clients arrive one at a time.

No opened facility can be closed in the future.

The connection between a client and a facility cannot be changed.

The edges of the tree cannot be removed.

Online Connected Facility Location Problem

In the online version of the Connected Facility Location problem the clients arrive one at a time.

No opened facility can be closed in the future.

The connection between a client and a facility cannot be changed.

The edges of the tree cannot be removed.

Online Connected Facility Location Problem

In the online version of the Connected Facility Location problem the clients arrive one at a time.

No opened facility can be closed in the future.

The connection between a client and a facility cannot be changed.

The edges of the tree cannot be removed.

Online Connected Facility Location Problem

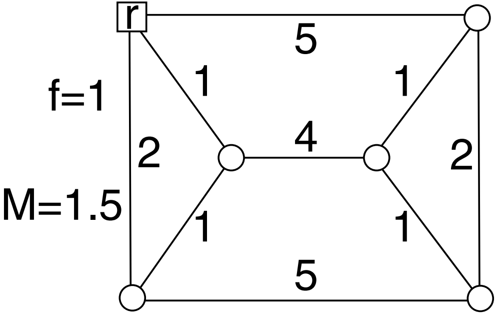
In the online version of the Connected Facility Location problem the clients arrive one at a time.

No opened facility can be closed in the future.

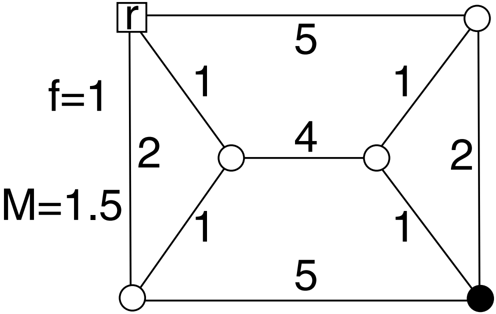
The connection between a client and a facility cannot be changed.

The edges of the tree cannot be removed.

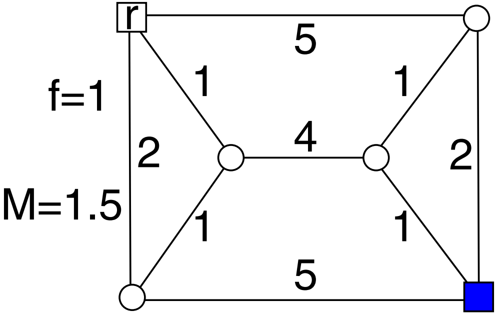
Online Connected Facility Location (ex.)



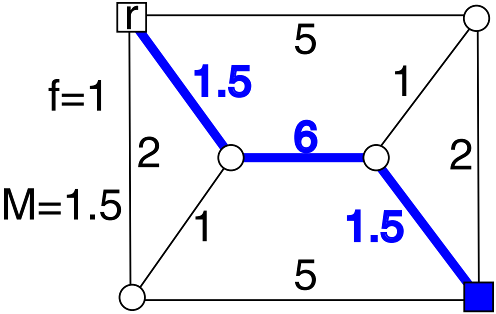
Online Connected Facility Location (ex.)



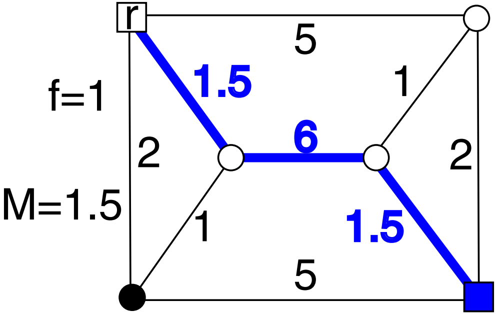
Online Connected Facility Location (ex.)



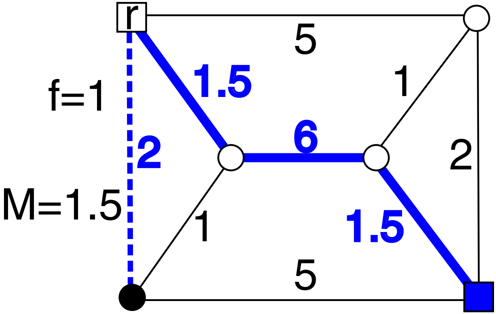
Online Connected Facility Location (ex.)



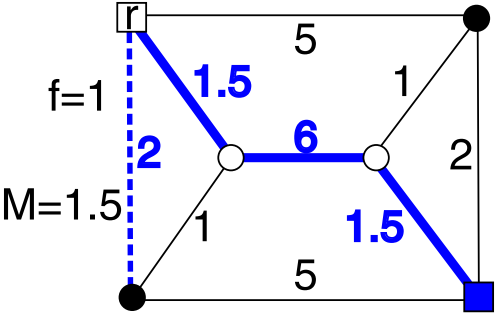
Online Connected Facility Location (ex.)



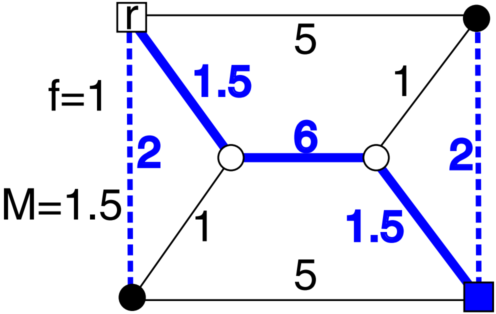
Online Connected Facility Location (ex.)



Online Connected Facility Location (ex.)



Online Connected Facility Location (ex.)



Lower Bound for the Online CFL Problem

The Online Steiner Tree problem can be reduced to the Online Connected Facility Location problem, by choosing all facility costs to be equal zero and $M = 1$.

There is a $\Omega(\log n)$ lower bound to the competitive ratio of any algorithm to the Online Steiner problem.

So the same bound applies to the competitive ratio of algorithms for the Online Connected Facility Location problem.

Lower Bound for the Online CFL Problem

The Online Steiner Tree problem can be reduced to the Online Connected Facility Location problem, by choosing all facility costs to be equal zero and $M = 1$.

There is a $\Omega(\log n)$ lower bound to the competitive ratio of any algorithm to the Online Steiner problem.

So the same bound applies to the competitive ratio of algorithms for the Online Connected Facility Location problem.

Lower Bound for the Online CFL Problem

The Online Steiner Tree problem can be reduced to the Online Connected Facility Location problem, by choosing all facility costs to be equal zero and $M = 1$.

There is a $\Omega(\log n)$ lower bound to the competitive ratio of any algorithm to the Online Steiner problem.

So the same bound applies to the competitive ratio of algorithms for the Online Connected Facility Location problem.

Online CFL Algorithm

Following we present a randomized algorithm for the Online Connected Facility Location problem.

This algorithm is based in the algorithm for the CFL due to Eisenbrand et al.

Online CFL Algorithm

Following we present a randomized algorithm for the Online Connected Facility Location problem.

This algorithm is based in the algorithm for the CFL due to Eisenbrand et al.

Algorithm 1: The Online CFL algorithm.

```
Input: G = (V, E), d, f, F, root r and M
send r to compFL as its first client;
while a new client j arrives do
    send j to compFL;
    include j in D^m with probability 1/M;
    if j ∈ D^m then
        T ← T ∪ {path(j, V(T))}; /* Core Tree */
        if v(j) is not opened then
            F^a ← F^a ∪ {v(j)}; /* Open Facility */
            T ← T ∪ {(v(j), j)}; /* Extension Tree */
        end
    end
    choose i ∈ F^a that is closest to j;
    D ← D ∪ {j}; a(j) ← i; /* Client Connection */
end
return (F^a \ {r}, T, a);
```

Analysis of the Online CFL Algorithm

We divide the algorithm cost between facilities opening cost (O), clients connection cost (C) and Steiner tree cost (S):

$$ALG_{OCFL}(D) = O + C + S.$$

We also divide the cost of the offline optimal solution in this way:

$$OPT_{CFL}(D) = O^* + C^* + S^*.$$

Analysis of the Online CFL Algorithm

We divide the algorithm cost between facilities opening cost (O), clients connection cost (C) and Steiner tree cost (S):

$$ALG_{OCFL}(D) = O + C + S.$$

We also divide the cost of the offline optimal solution in this way:

$$OPT_{CFL}(D) = O^* + C^* + S^*.$$

Analysis of the Online CFL Algorithm (cont.)

The compFL algorithm uses a nonnegative dual variable α_j associated with each client j.

Lemma (α properties)

$$O_{\mathrm{compFL}}(D) \le \sum_{j \in D} \alpha_j \ ,$$

$$C_{\mathrm{compFL}}(D) \le \sum_{j \in D} \alpha_j \ ,$$

$$2 \sum_{j \in D} \alpha_j \le c_{\mathrm{OFL}} \mathrm{OPT}_{\mathrm{FL}}(D) \ ,$$

$$\alpha_j \ge d(j, i) \ .$$

Analysis of the Online CFL Algorithm (cont.)

The compFL algorithm uses a nonnegative dual variable α_j associated with each client j.

Lemma (α properties)

$$O_{\mathrm{compFL}}(D) \leq \sum_{j \in D} \alpha_j \ ,$$

$$C_{\mathrm{compFL}}(D) \leq \sum_{j \in D} \alpha_j \ ,$$

$$2 \sum_{j \in D} \alpha_j \leq c_{\mathrm{OFL}} \mathrm{OPT}_{\mathrm{FL}}(D) \ ,$$

$$\alpha_j \geq d(j, i) \ .$$

Analysis of the Online CFL Algorithm (cont.)

Remember that

$$\mathrm{compFL}(r + D) = O_{\mathrm{compFL}}(r + D) + C_{\mathrm{compFL}}(r + D) \ .$$

Lemma (compFL bound)

$$O_{\mathrm{compFL}}(r + D) \leq \frac{1}{2} c_{\mathrm{OFL}}(O^*(D) + C^*(D)) \ ,$$

$$C_{\mathrm{compFL}}(r + D) \leq \frac{1}{2} c_{\mathrm{OFL}}(O^*(D) + C^*(D)) \ .$$

Analysis of the Online CFL Algorithm (cont.)

Remember that

$$\text{compFL}(r + D) = O_{\text{compFL}}(r + D) + C_{\text{compFL}}(r + D) \ .$$

Lemma (compFL bound)

$$O_{\text{compFL}}(r + D) \leq \frac{1}{2} c_{\text{OFL}}(O^*(D) + C^*(D)) \ ,$$

$$C_{\text{compFL}}(r + D) \leq \frac{1}{2} c_{\text{OFL}}(O^*(D) + C^*(D)) \ .$$

Analysis of the Online CFL Algorithm (cont.)

Let $S_{\text{core}}(D)$ be the cost of the core tree.

Lemma (Core tree bound)

$$E[S_{\text{core}}(D)] \leq c_{\text{OST}}(S^*(D) + C^*(D)) \ .$$

Analysis of the Online CFL Algorithm (cont.)

Let $S_{\text{core}}(D)$ be the cost of the core tree.

Lemma (Core tree bound)

$$E[S_{\text{core}}(D)] \leq c_{\text{OST}}(S^*(D) + C^*(D)) \ .$$

Analysis of the Online CFL Algorithm (cont.)

Let $S_{\mathrm{ext}}(D)$ be the cost of the tree extensions.

Lemma (Tree extensions bound)

$$E[S_{\mathrm{ext}}(D)] \leq C_{\mathrm{compFL}}(r + D) \ .$$

Using the two previous lemmas we bound the expected Steiner tree cost $S(D)$ of the Online CFL algorithm.

Analysis of the Online CFL Algorithm (cont.)

Let $S_{\text{ext}}(D)$ be the cost of the tree extensions.

Lemma (Tree extensions bound)

$$E[S_{\text{ext}}(D)] \leq C_{\text{compFL}}(r + D) \ .$$

Using the two previous lemmas we bound the expected Steiner tree cost $S(D)$ of the Online CFL algorithm.

Analysis of the Online CFL Algorithm (cont.)

Let $S_{\text{ext}}(D)$ be the cost of the tree extensions.

Lemma (Tree extensions bound)

$$E[S_{\text{ext}}(D)] \leq C_{\text{compFL}}(r + D) \ .$$

Using the two previous lemmas we bound the expected Steiner tree cost $S(D)$ of the Online CFL algorithm.

Analysis of the Online CFL Algorithm (cont.)

Now we bound the expected client connection cost $C(D)$.

For each marked client j', we keep a set $N(j')$ of clients called the *neighborhood* of j'.

A client j is added to $N(j')$ if j' is the marked client that is closest to j, and if j satisfies

$$d(j, j') + \alpha_j < \frac{1}{3} d(j, F^a_{n(j)}) \ .$$

Analysis of the Online CFL Algorithm (cont.)

Now we bound the expected client connection cost $C(D)$.

For each marked client j', we keep a set $N(j')$ of clients called the *neighborhood* of j'.

A client j is added to $N(j')$ if j' is the marked client that is closest to j, and if j satisfies

$$d(j, j') + \alpha_j < \frac{1}{3} d(j, F^a_{n(j)}) \ .$$

Analysis of the Online CFL Algorithm (cont.)

Now we bound the expected client connection cost $C(D)$.

For each marked client j', we keep a set $N(j')$ of clients called the *neighborhood* of j'.

A client j is added to $N(j')$ if j' is the marked client that is closest to j, and if j satisfies

$$d(j, j') + \alpha_j < \frac{1}{3} d(j, F^a_{n(j)}) \ .$$

Analysis of the Online CFL Algorithm (cont.)

We call the clients that are in some neighborhood by neighbors and denote them by D^N.

The other clients we call non-neighbors and denote by $\overline{D^N}$.

Lemma (Non-neighbors connection bound)

$$E[C(\overline{D^N})] \leq \frac{3}{2} c_{\mathrm{OFL}}(O^*(D) + C^*(D)) + 3E\left[\sum_{j \in \overline{D^N}} d(j, D^m_{n(j)})\right].$$

Analysis of the Online CFL Algorithm (cont.)

We call the clients that are in some neighborhood by neighbors and denote them by D^N.

The other clients we call non-neighbors and denote by $\overline{D^N}$.

Lemma (Non-neighbors connection bound)

$$E[C(\overline{D^N})] \leq \frac{3}{2} c_{\mathrm{OFL}}(O^*(D) + C^*(D)) + 3E\left[\sum_{j \in \overline{D^N}} d(j, D^m_{n(j)})\right].$$

Analysis of the Online CFL Algorithm (cont.)

We call the clients that are in some neighborhood by neighbors and denote them by D^N.

The other clients we call non-neighbors and denote by $\overline{D^N}$.

Lemma (Non-neighbors connection bound)

$$E[C(\overline{D^N})] \leq \frac{3}{2} c_{\mathrm{OFL}}(O^*(D) + C^*(D)) + 3E\left[\sum_{j \in \overline{D^N}} d(j, D^m_{n(j)})\right] .$$

Analysis of the Online CFL Algorithm (cont.)

We may split the neighborhood $N(j')$ of a marked client j' into phases.

A phase k ends when the algorithm opens a facility i, that satisfies

$$d(j', i) < \frac{1}{2} d(j', p_k(j')) \ . \tag{0.1}$$

Lemma (Neighbors' facility closeness)

For any $j' \in D^m$, $k \in \mathrm{phase}(j')$, and $j \in N_k(j')$ we have that

$$d(j', v(j)) < \frac{1}{2} d(j', p_k(j')) \ .$$

Analysis of the Online CFL Algorithm (cont.)

We may split the neighborhood $N(j')$ of a marked client j' into phases.

A phase k ends when the algorithm opens a facility i, that satisfies

$$d(j', i) < \frac{1}{2} d(j', p_k(j')) \ . \tag{0.1}$$

Lemma (Neighbors' facility closeness)

For any $j' \in D^m$, $k \in \mathrm{phase}(j')$, and $j \in N_k(j')$ we have that

$$d(j', v(j)) < \frac{1}{2} d(j', p_k(j')) \ .$$

Analysis of the Online CFL Algorithm (cont.)

We may split the neighborhood $N(j')$ of a marked client j' into phases.

A phase k ends when the algorithm opens a facility i, that satisfies

$$d(j', i) < \frac{1}{2} d(j', p_k(j')) \ . \tag{0.1}$$

Lemma (Neighbors' facility closeness)

For any $j' \in D^m$, $k \in \mathrm{phase}(j')$, and $j \in N_k(j')$ we have that

$$d(j', v(j)) < \frac{1}{2} d(j', p_k(j')) \ .$$

Analysis of the Online CFL Algorithm (cont.)

A geometric distribution is a random variable that performs a sequence of independent trials until the first success.

We bound the expected number of clients in a phase neighborhood using a geometric distribution.

Lemma (Phase length bound)

For any $j' \in D^m$ and $k \in \mathrm{phase}(j')$, we have

$$E[|N_k(j')|] \leq M \quad .$$

Analysis of the Online CFL Algorithm (cont.)

A geometric distribution is a random variable that performs a sequence of independent trials until the first success.

We bound the expected number of clients in a phase neighborhood using a geometric distribution.

Lemma (Phase length bound)

For any $j' \in D^m$ and $k \in \mathrm{phase}(j')$, we have

$$E[|N_k(j')|] \leq M \quad .$$

Analysis of the Online CFL Algorithm (cont.)

A geometric distribution is a random variable that performs a sequence of independent trials until the first success.

We bound the expected number of clients in a phase neighborhood using a geometric distribution.

Lemma (Phase length bound)

For any $j' \in D^m$ and $k \in \mathrm{phase}(j')$, we have

$$E[|N_k(j')|] \leq M \ .$$

Analysis of the Online CFL Algorithm (cont.)

Using the previous lemma we bound the expected connection cost of the clients in D^N.

Lemma (Neighbors connection bound)

$$E[C(D^N)] \leq E\left[\sum_{j \in D^N} d(j, D^m_{n(j)}))\right] + 2C_{\mathrm{compFL}}(r + D) \ .$$

Analysis of the Online CFL Algorithm (cont.)

Using the previous lemma we bound the expected connection cost of the clients in D^N.

Lemma (Neighbors connection bound)

$$E[C(D^N)] \leq E\left[\sum_{j \in D^N} d(j, D^m_{n(j)})\right] + 2C_{\mathrm{compFL}}(r + D) \ .$$

Analysis of the Online CFL Algorithm (cont.)

Now we prove an auxiliary lemma.

Lemma (Auxiliary connection bound)

$$E\left[\sum_{j \in D} d(j, D^m_{n(j)})\right] \leq c_{\mathrm{OST}}(S^*(D) + C^*(D)) \ .$$

Using the previous lemmas and that the competitive ratio of compFL and compST is $\mathrm{O}(\log n)$, we prove our main result in the next theorem.

Analysis of the Online CFL Algorithm (cont.)

Now we prove an auxiliary lemma.

Lemma (Auxiliary connection bound)

$$E\left[\sum_{j\in D} d(j, D^m_{n(j)})\right] \leq c_{\mathrm{OST}}(S^*(D) + C^*(D)) \ .$$

Using the previous lemmas and that the competitive ratio of compFL and compST is $\mathrm{O}(\log n)$, we prove our main result in the next theorem.

Analysis of the Online CFL Algorithm (cont.)

Now we prove an auxiliary lemma.

Lemma (Auxiliary connection bound)

$$E\left[\sum_{j\in D} d(j, D^m_{n(j)})\right] \leq \mathrm{cost}(S^*(D) + C^*(D)) \ .$$

Using the previous lemmas and that the competitive ratio of compFL and compST is $\mathrm{O}(\log n)$, we prove our main result in the next theorem.

Analysis of the Online CFL Algorithm (cont.)

Theorem

$E[\mathrm{ALG}_{\mathrm{OCFL}}(D)] = \mathrm{O}(\log n)\mathrm{OPT}_{\mathrm{CFL}}(D)$.

Demonstração.

$$\begin{aligned}
E[\mathrm{ALG}_{\mathrm{OCFL}}(D)] &= E[O(D) + S(D) + C(D)] \\
&\leq E\Big[O_{\mathrm{compFL}}(r + D) \\
&\quad + (S_{\mathrm{core}}(D) + S_{\mathrm{ext}}(D)) \\
&\quad + \Big(C(\overline{D^N}) + C(D^N)\Big)\Big] \\
&= \mathrm{O}(\log n)\mathrm{OPT}_{\mathrm{CFL}}(D) \quad ,
\end{aligned}$$

Analysis of the Online CFL Algorithm (cont.)

Theorem

$E[\mathrm{ALG}_{\mathrm{OCFL}}(D)] = \mathrm{O}(\log n)\mathrm{OPT}_{\mathrm{CFL}}(D)$.

Demonstração.

$$
\begin{aligned}
E[\mathrm{ALG}_{\mathrm{OCFL}}(D)] &= E[O(D) + S(D) + C(D)] \\
&\leq E\Big[O_{\mathrm{compFL}}(r + D) \\
&\quad + (S_{\mathrm{core}}(D) + S_{\mathrm{ext}}(D)) \\
&\quad + \left(C(\overline{D^N}) + C(D^N)\right)\Big] \\
&= \mathrm{O}(\log n)\mathrm{OPT}_{\mathrm{CFL}}(D) \quad ,
\end{aligned}
$$

■

References

M. Imase and B.M. Waxman.
Dynamic Steiner Tree Problem.
SIAM Journal on Discrete Mathematics 4(3), 369–384 (1991)

A. Gupta, A. Kumar, M. Pál and T. Roughgarden.
Approximation via cost sharing: Simpler and better approximation algorithms for network design.
Journal of the ACM 54(3), Article 11 (2007)

D. Fotakis.
A Primal-Dual Algorithm for Online Non-Uniform Facility Location.
Journal of Discrete Algorithms 5(1), 141–148 (2007)

References (cont.)

- C. Nagarajan and D.P. Williamson.
  Offline and Online Facility Leasing.
  Discrete Optimization 10(4), 361–370 (2013)

- F. Eisenbrand, F. Grandoni, T. Rothvoß and G. Schäfer.
  Connected Facility Location Via Random Facility Sampling and Core Detouring
  Journal of Computer and System Sciences 76(8), 709–726 (2010)

- M.C. San Felice, D.P. Williamson, O. Lee.
  The Online Connected Facility Location Problem
  In: LATIN 2014, LNCS 8392, 574–585, Springer-Verlag (2014)

Acknowledgements

Thank you!

Questions?

Acknowledgements

Thank you!

Questions?