The Online Connected Facility Location Problem[2]

Mário César San Felice
Advisor: Orlando Lee
Supervisor Abroad: David P. Williamson

UNICAMP
Institute of Computing
Laboratory of Optimization and Combinatorics

UNICAMP

December 13th, 2013

[2]grant No. 2010/15535-1, São Paulo Research Foundation (FAPESP)

Combinatorial Optimization Problems

Maximization or minimization problems in which, for each input there is a set of feasible solutions and, for each solution there is a cost associated with it.

As an example, lets take the Steiner Tree problem.

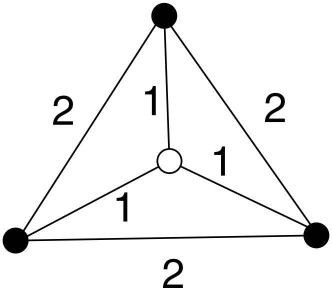
Combinatorial Optimization Problems

Maximization or minimization problems in which, for each input there is a set of feasible solutions and, for each solution there is a cost associated with it.

As an example, lets take the Steiner Tree problem.

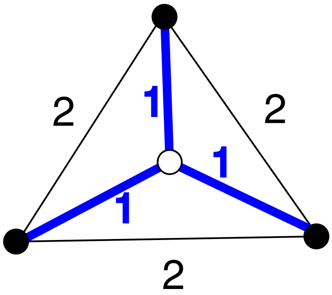
Steiner Tree Problem

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.



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.

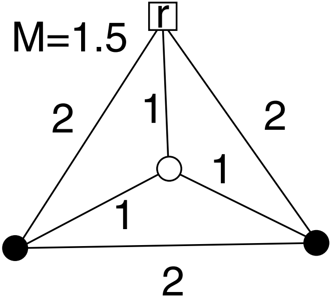
Rent-or-Buy Problems

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.

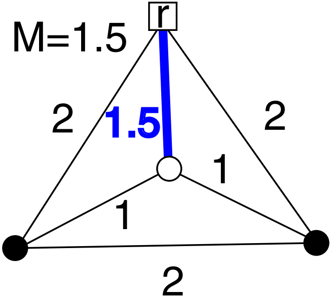
Single Source Rent-or-Buy Problem

A rent-or-buy version of the Steiner Tree problem in which all terminals must be connected to a source. The algorithm can decide between renting and 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.



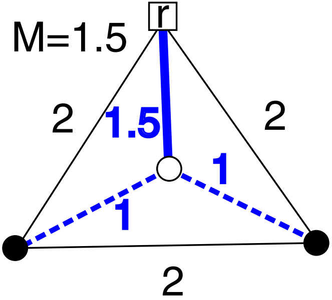
Single Source Rent-or-Buy Problem

A rent-or-buy version of the Steiner Tree problem in which all terminals must be connected to a source. The algorithm can decide between renting and 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. The algorithm can decide between renting and 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.



The Sample-and-Augment Technique

A randomized technique to design algorithms for rent-or-buy problems. The central idea is to decide between renting or buying a resource using a coin toss. The buying probability is greater the least is the ratio between buying and renting costs.

As an example, lets take a sample-and-augment algorithm for the Single Source Rent-or-Buy problem.

The Sample-and-Augment Technique

A randomized technique to design algorithms for rent-or-buy problems. The central idea is to decide between renting or buying a resource using a coin toss. The buying probability is greater the least is the ratio between buying and renting costs.

As an example, lets take a sample-and-augment algorithm for the Single Source Rent-or-Buy problem.

A Sample-and-Augment Algorithm

Algorithm 1: The Sample-and-Augment SSRoB Algorithm

Mark each terminal with probability $\frac{1}{M}$.

Find a tree T for the marked terminals using an approximation algorithm for the Steiner Tree problem and buy this tree.

Connect the remaining terminals to T using rented shortest paths.

This algorithm has a constant approximation ratio to the Single Source Rent-or-Buy problem.

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.

As an example, lets take the Online Steiner Tree problem.

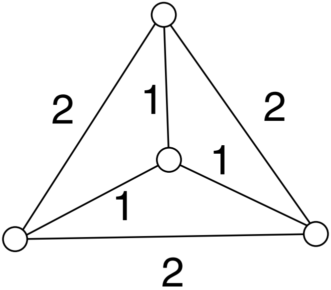
Online Problems

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.

As an example, lets take the Online Steiner Tree problem.

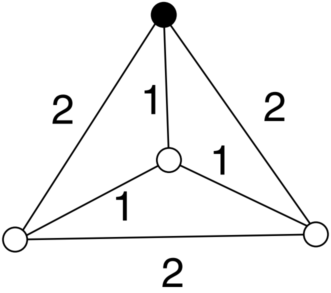
Online Steiner Tree Problem

This problem is defined similarly to the Steiner Tree problem, except that the terminal nodes arrive one at a time. At all times the terminals must be connected by a tree and no edge used may be removed in the future.



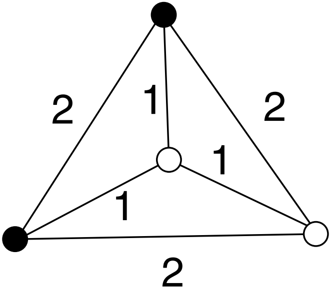
Online Steiner Tree Problem

This problem is defined similarly to the Steiner Tree problem, except that the terminal nodes arrive one at a time. At all times the terminals must be connected by a tree and no edge used may be removed in the future.



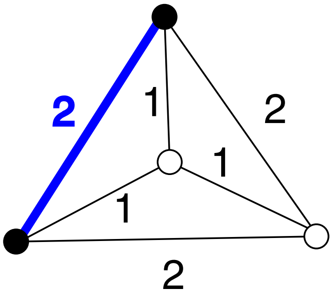
Online Steiner Tree Problem

This problem is defined similarly to the Steiner Tree problem, except that the terminal nodes arrive one at a time. At all times the terminals must be connected by a tree and no edge used may be removed in the future.



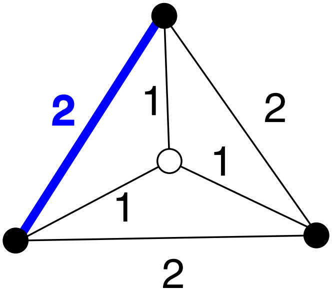
Online Steiner Tree Problem

This problem is defined similarly to the Steiner Tree problem, except that the terminal nodes arrive one at a time. At all times the terminals must be connected by a tree and no edge used may be removed in the future.



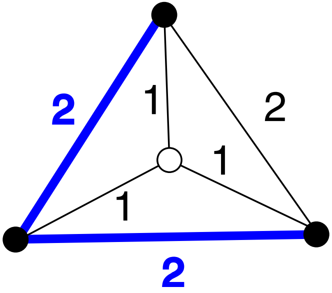
Online Steiner Tree Problem

This problem is defined similarly to the Steiner Tree problem, except that the terminal nodes arrive one at a time. 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. At all times the terminals must be connected by a tree and no edge used may be removed 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:

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

There are $\mathrm{O}(\log n)$-competitive algorithms for the Online Steiner Tree problem. Also, it is known a $\Omega(\log n)$ lower bound to the competitive ratio of any algorithm to this problem.

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:

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

There are $\mathrm{O}(\log n)$-competitive algorithms for the Online Steiner Tree problem. Also, it is known a $\Omega(\log n)$ lower bound to the competitive ratio of any algorithm to this problem.

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:

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

There are $\mathrm{O}(\log n)$-competitive algorithms for the Online Steiner Tree problem. Also, it is known a $\Omega(\log n)$ lower bound to the competitive ratio of any algorithm to this problem.

Online Rent-or-Buy Problems

It is possible to define a problem that has both rent-or-buy and online characteristics.

As an example, in the Online Single Source Rent-or-Buy problem the terminals arrive one at a time and, for each one that arrives the algorithm has to connect it to the source by renting or buying edges.

There is a sample-and-augment algorithm known to this problem that is $O(\log n)$-competitive.

Online Rent-or-Buy Problems

It is possible to define a problem that has both rent-or-buy and online characteristics.

As an example, in the Online Single Source Rent-or-Buy problem the terminals arrive one at a time and, for each one that arrives the algorithm has to connect it to the source by renting or buying edges.

There is a sample-and-augment algorithm known to this problem that is $O(\log n)$-competitive.

Online Rent-or-Buy Problems

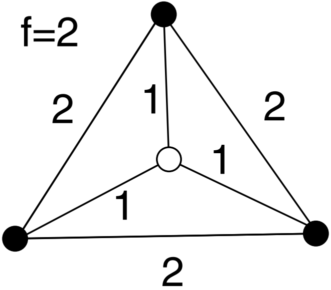
It is possible to define a problem that has both rent-or-buy and online characteristics.

As an example, in the Online Single Source Rent-or-Buy problem the terminals arrive one at a time and, for each one that arrives the algorithm has to connect it to the source by renting or buying edges.

There is a sample-and-augment algorithm known to this problem that is $O(\log n)$-competitive.

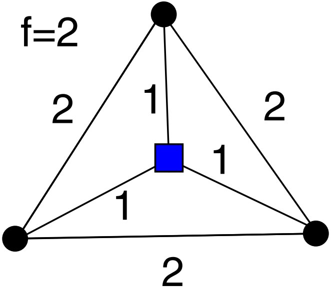
The Facility Location Problem

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).



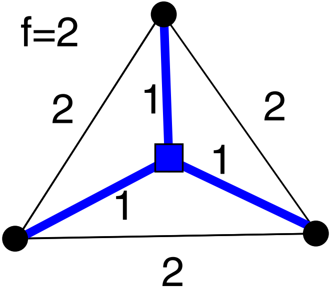
The Facility Location Problem

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).



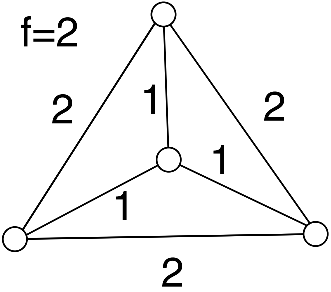
The Facility Location Problem

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).



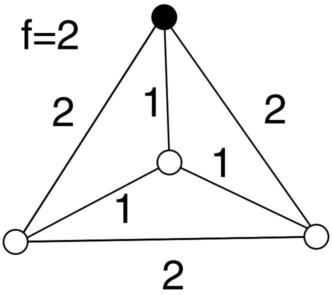
The Online Facility Location Problem

In the online version of the Facility Location problem the clients arrive one at a time and no opened facility can be closed in the future, nor the connection between a client and a facility can be changed.



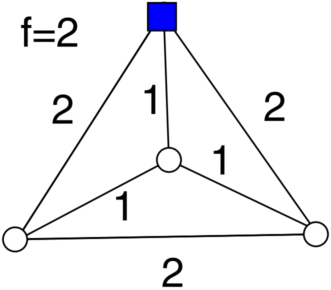
The Online Facility Location Problem

In the online version of the Facility Location problem the clients arrive one at a time and no opened facility can be closed in the future, nor the connection between a client and a facility can be changed.



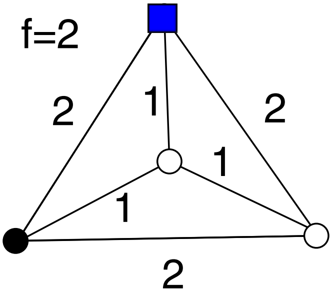
The Online Facility Location Problem

In the online version of the Facility Location problem the clients arrive one at a time and no opened facility can be closed in the future, nor the connection between a client and a facility can be changed.



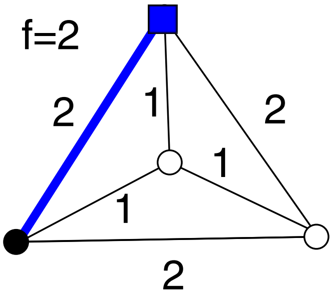
The Online Facility Location Problem

In the online version of the Facility Location problem the clients arrive one at a time and no opened facility can be closed in the future, nor the connection between a client and a facility can be changed.



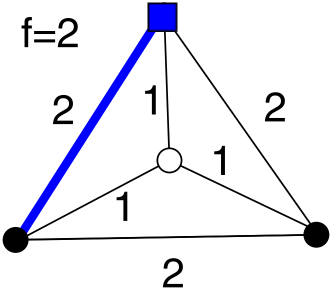
The Online Facility Location Problem

In the online version of the Facility Location problem the clients arrive one at a time and no opened facility can be closed in the future, nor the connection between a client and a facility can be changed.



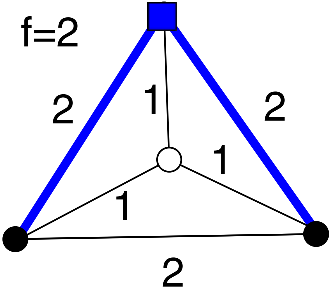
The Online Facility Location Problem

In the online version of the Facility Location problem the clients arrive one at a time and 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 and 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 (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 to 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 to 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 need 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.

$$\sum_{j \in D} d(j, F') + \sum_{i \in F'} f(i) + M \sum_{e \in T} d(e)$$

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 need 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.

$$\sum_{j \in D} d(j, F') + \sum_{i \in F'} f(i) + M \sum_{e \in T} d(e)$$

Connected Facility Location Problem

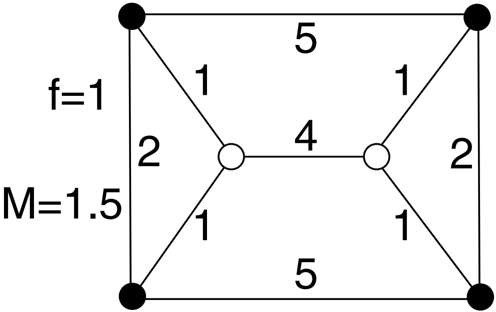
This problem is a combination of the Facility Location problem with the Steiner Tree problem.

There is a set of clients that need 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.

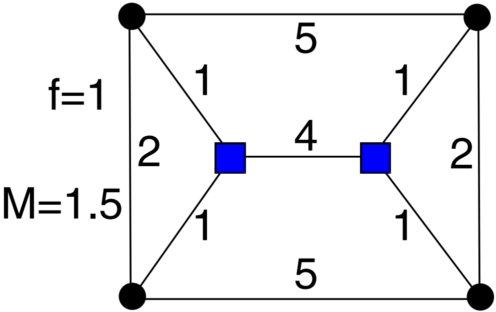
The goal is to minimize the total cost of connecting clients, opening facilities and building the tree.

$$\sum_{j \in D} d(j, F') + \sum_{i \in F'} f(i) + M \sum_{e \in T} d(e)$$

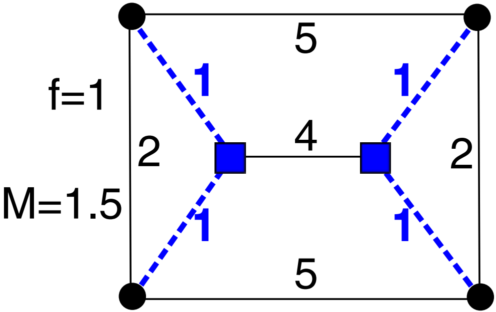
Connected Facility Location Problem (cont.)



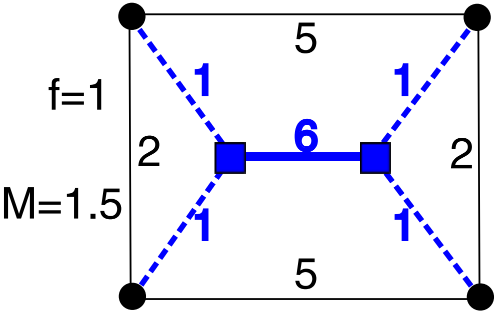
Connected Facility Location Problem (cont.)



Connected Facility Location Problem (cont.)

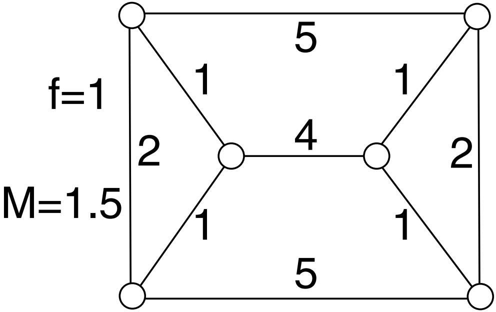


Connected Facility Location Problem (cont.)



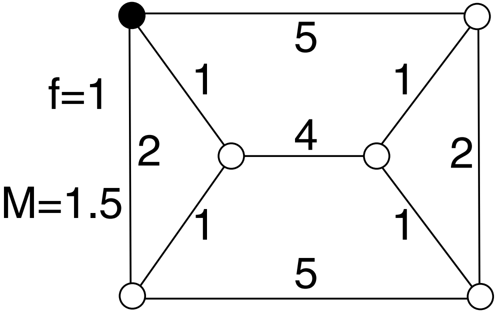
Online Connected Facility Location Problem

In the online version of the Connected Facility Location problem the clients arrive one at a time. Also, no opened facility can be closed in the future, the connection between a client and a facility cannot be changed and the edges of the tree cannot be removed.



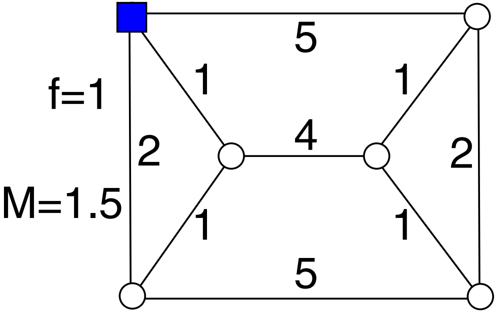
Online Connected Facility Location Problem

In the online version of the Connected Facility Location problem the clients arrive one at a time. Also, no opened facility can be closed in the future, the connection between a client and a facility cannot be changed and the edges of the tree cannot be removed.



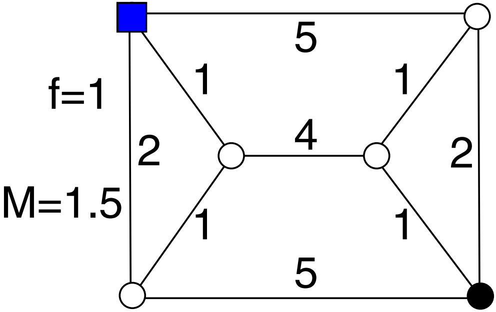
Online Connected Facility Location Problem

In the online version of the Connected Facility Location problem the clients arrive one at a time. Also, no opened facility can be closed in the future, the connection between a client and a facility cannot be changed and the edges of the tree cannot be removed.



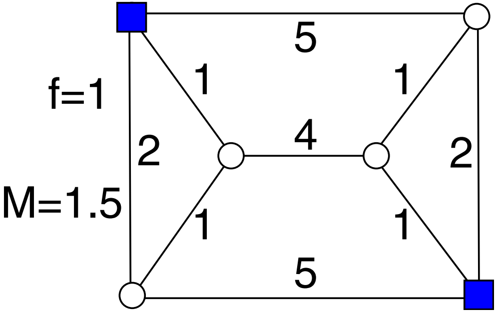
Online Connected Facility Location Problem

In the online version of the Connected Facility Location problem the clients arrive one at a time. Also, no opened facility can be closed in the future, the connection between a client and a facility cannot be changed and the edges of the tree cannot be removed.



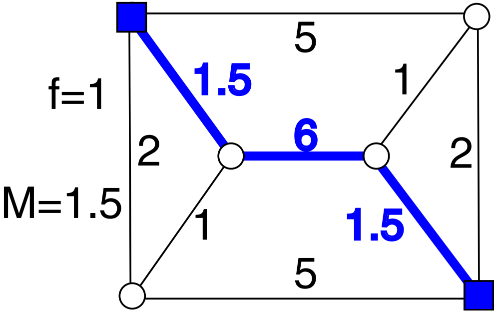
Online Connected Facility Location Problem

In the online version of the Connected Facility Location problem the clients arrive one at a time. Also, no opened facility can be closed in the future, the connection between a client and a facility cannot be changed and the edges of the tree cannot be removed.



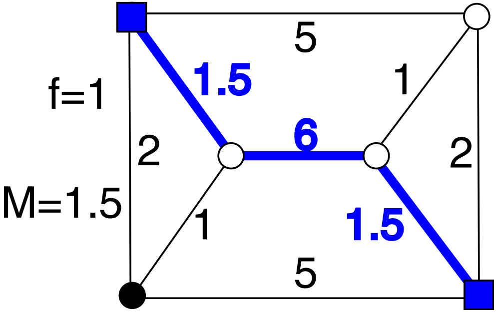
Online Connected Facility Location Problem

In the online version of the Connected Facility Location problem the clients arrive one at a time. Also, no opened facility can be closed in the future, the connection between a client and a facility cannot be changed and the edges of the tree cannot be removed.



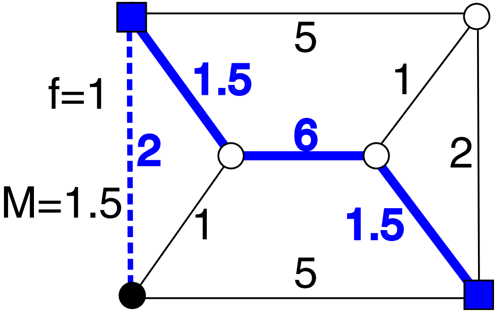
Online Connected Facility Location Problem

In the online version of the Connected Facility Location problem the clients arrive one at a time. Also, no opened facility can be closed in the future, the connection between a client and a facility cannot be changed and the edges of the tree cannot be removed.



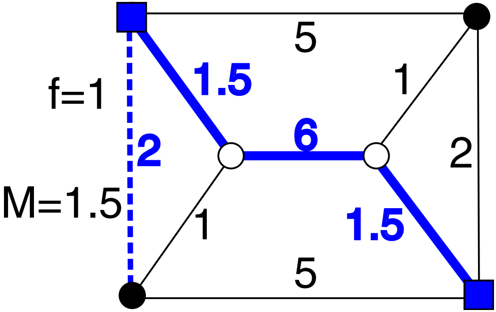
Online Connected Facility Location Problem

In the online version of the Connected Facility Location problem the clients arrive one at a time. Also, no opened facility can be closed in the future, the connection between a client and a facility cannot be changed and the edges of the tree cannot be removed.



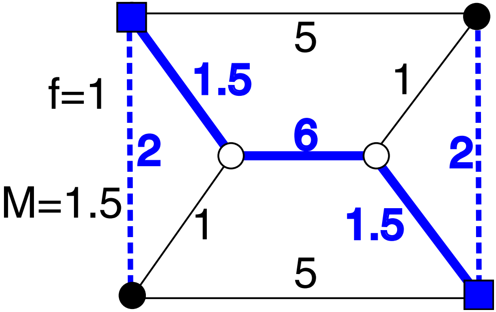
Online Connected Facility Location Problem

In the online version of the Connected Facility Location problem the clients arrive one at a time. Also, no opened facility can be closed in the future, the connection between a client and a facility cannot be changed and 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. Also, no opened facility can be closed in the future, the connection between a client and a facility cannot be changed and the edges of the tree cannot be removed.



Online CFL Algorithm

Following we present a sample-and-augment algorithm for the Online Connected Facility Location problem. This algorithm is based in the algorithm for the CFL due to Eisenbrand et al.

Notice that, while the Online Connected Facility Location problem is not a rent-or-buy problem, it has some characteristics that allow us to use the sample-and-augment technique to design an algorithm for it.

Online CFL Algorithm

Following we present a sample-and-augment algorithm for the Online Connected Facility Location problem. This algorithm is based in the algorithm for the CFL due to Eisenbrand et al.

Notice that, while the Online Connected Facility Location problem is not a rent-or-buy problem, it has some characteristics that allow us to use the sample-and-augment technique to design an algorithm for it.

Algorithm 2: The Online CFL algorithm.

Data: $G = (V, E)$, d, f, F, root r and M

$D \leftarrow \emptyset$; $F' \leftarrow \emptyset$; $T \leftarrow \emptyset$;

$f(r) \leftarrow 0$;

send r to compFL;

$F' \leftarrow F' \cup \{r\}$; $V(T) \leftarrow V(T) \cup \{r\}$;

while *a new client j arrives* **do**

    send j to compFL;

    sample j with probability $p = \frac{1}{M}$;

    if *j was sampled and connected to a facility i that wasn't open* **then**

        $F' \leftarrow F' \cup \{i\}$;

        $T \leftarrow T \cup \{(i, j)\} \cup \{path(j, V(T))\}$;

    end

    let i be the closest open facility to j;

    $D \leftarrow D \cup \{j\}$; $a(j) \leftarrow i$;

end

return $(F' \setminus \{r\}, T, a)$;

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.)

Lemma (Opening Cost)

$O \leq c_{\mathrm{OFL}}(O^* + C^*)$.

Demonstração.

Let O_{compFL} be the facility opening cost paid by compFL to serve $\{r\} \cup D$. Once our algorithm opens a subset of the facilities opened by compFL to serve $\{r\} \cup D$ we have that:

$$O \leq O_{\mathrm{compFL}} \leq c_{\mathrm{OFL}} \mathrm{OPT}_{\mathrm{FL}}(\{r\} \cup D) \leq c_{\mathrm{OFL}}(O^* + C^*) \ ,$$

where the last inequality follows since the optimal solution for CFL is a feasible solution for the OFL. ∎

Analysis of the Online CFL Algorithm (cont.)

Lemma (Opening Cost)

$O \leq c_{\mathrm{OFL}}(O^* + C^*)$.

Demonstração.

Let O_{compFL} be the facility opening cost paid by compFL to serve $\{r\} \cup D$. Once our algorithm opens a subset of the facilities opened by compFL to serve $\{r\} \cup D$ we have that:

$$O \quad \leq \quad O_{\mathrm{compFL}} \leq c_{\mathrm{OFL}} \mathrm{OPT}_{\mathrm{FL}}(\{r\} \cup D) \leq c_{\mathrm{OFL}}(O^* + C^*) \quad ,$$

where the last inequality follows since the optimal solution for CFL is a feasible solution for the OFL. ■

Analysis of the Online CFL Algorithm (cont.)

Lemma (Steiner Cost)

$E[S] \leq c_{\mathrm{OST}}(S^* + C^*) + c_{\mathrm{OFL}}(O^* + C^*)$.

Idea: The Online CFL algorithm builds a tree connecting the root r to each client in D''. Then it augments T connecting each client $j \in D''$ to the facility i that was opened by it. So:

$$\begin{aligned} S &\leq M \mathrm{compST}(\{r\} \cup D'') + M \sum_{j \in \{r\} \cup D''} d(j, a(j)) \\ &\leq M c_{\mathrm{OST}} \mathrm{OPT}_{\mathrm{ST}}(\{r\} \cup D') + M \sum_{j \in D'} d(j, a(j)) \ . \end{aligned}$$

Analysis of the Online CFL Algorithm (cont.)

Lemma (Steiner Cost)

$E[S] \leq c_{\mathrm{OST}}(S^* + C^*) + c_{\mathrm{OFL}}(O^* + C^*)$.

Idea: The Online CFL algorithm builds a tree connecting the root r to each client in D''. Then it augments T connecting each client $j \in D''$ to the facility i that was opened by it. So:

$$\begin{aligned} S &\leq M\mathrm{compST}(\{r\} \cup D'') + M \sum_{j \in \{r\} \cup D''} d(j, a(j)) \\ &\leq M c_{\mathrm{OST}} \mathrm{OPT}_{\mathrm{ST}}(\{r\} \cup D') + M \sum_{j \in D'} d(j, a(j)) \ . \end{aligned}$$

Analysis of the Online CFL Algorithm (cont.)

$$\begin{aligned} E[\mathrm{OPT}_{\mathrm{ST}}(\{r\} \cup D')] &\leq E\left[\frac{S^*}{M}\right] + E\left[\sum_{j \in D'} d(j, a^*(j))\right] \\ &\leq \frac{S^*}{M} + \sum_{j \in D} \frac{1}{M} d(j, a^*(j)) \leq \frac{S^*}{M} + \frac{C^*}{M} \ . \end{aligned}$$

Lemma (Connection Cost)

$E[C] \leq c_{\mathrm{OFL}}(O^* + C^*) + c_{\mathrm{OST}}(S^* + C^* + c_{\mathrm{OFL}}(O^* + C^*))$.

Analysis of the Online CFL Algorithm (cont.)

$$
\begin{aligned}
E[\mathrm{OPT}_{\mathrm{ST}}(\{r\} \cup D')] &\leq E\left[\frac{S^*}{M}\right] + E\left[\sum_{j \in D'} d(j, a^*(j))\right] \\
&\leq \frac{S^*}{M} + \sum_{j \in D} \frac{1}{M} d(j, a^*(j)) \leq \frac{S^*}{M} + \frac{C^*}{M} .
\end{aligned}
$$

Lemma (Connection Cost)

$E[C] \leq c_{\mathrm{OFL}}(O^* + C^*) + c_{\mathrm{OST}}(S^* + C^* + c_{\mathrm{OFL}}(O^* + C^*)).$

Analysis of the Online CFL Algorithm (cont.)

Theorem

$E[\mathrm{ALG}_{\mathrm{OCFL}}(D)] \in \mathrm{O}(\log n^2 \mathrm{OPT}_{\mathrm{CFL}}(D))$.

Demonstração.

$$\begin{aligned} E[\mathrm{ALG}_{\mathrm{OCFL}}(D)] &= E[O + S + C] \\ &\leq c_{\mathrm{OFL}}(O^* + C^*) + (c_{\mathrm{OST}}(S^* + C^*) \\ &\quad + c_{\mathrm{OFL}}(O^* + C^*)) + (c_{\mathrm{OFL}}(O^* + C^*) \\ &\quad + c_{\mathrm{OST}}(S^* + C^* + c_{\mathrm{OFL}}(O^* + C^*))) \\ &= \mathrm{O}(\log^2 n)\mathrm{OPT}_{\mathrm{CFL}}(D) \ . \end{aligned}$$

where the last inequality follows because $c_{\mathrm{OFL}} \leq 4 \log n$ and $c_{\mathrm{OST}} \leq \log n$. ∎

Analysis of the Online CFL Algorithm (cont.)

Theorem

$E[\mathrm{ALG}_{\mathrm{OCFL}}(D)] \in \mathrm{O}(\log n^2 \mathrm{OPT}_{\mathrm{CFL}}(D))$.

Demonstração.

$$\begin{aligned} E[\mathrm{ALG}_{\mathrm{OCFL}}(D)] &= E[O + S + C] \\ &\leq c_{\mathrm{OFL}}(O^* + C^*) + (c_{\mathrm{OST}}(S^* + C^*) \\ &\quad + c_{\mathrm{OFL}}(O^* + C^*)) + (c_{\mathrm{OFL}}(O^* + C^*) \\ &\quad + c_{\mathrm{OST}}(S^* + C^* + c_{\mathrm{OFL}}(O^* + C^*))) \\ &= \mathrm{O}(\log^2 n)\mathrm{OPT}_{\mathrm{CFL}}(D) \ . \end{aligned}$$

where the last inequality follows because $c_{\mathrm{OFL}} \leq 4 \log n$ and $c_{\mathrm{OST}} \leq \log n$. ∎

Lower Bound for the Online CFL Problem

It is possible to reduce the Online Steiner Tree problem to the Online Connected Facility Location problem by choosing all facility costs to be equal zero and $M = 1$.

So, the $\Omega(\log n)$ lower bound to the competitive ratio of any algorithm to the Online Steiner problem also applies to algorithms to the Online Connected Facility Location problem.

Lower Bound for the Online CFL Problem

It is possible to reduce the Online Steiner Tree problem to the Online Connected Facility Location problem by choosing all facility costs to be equal zero and $M = 1$.

So, the $\Omega(\log n)$ lower bound to the competitive ratio of any algorithm to the Online Steiner problem also applies to algorithms to the Online Connected Facility Location problem.

References

M. Imase and M.B. Waxman.
Dynamic Steiner Tree Problem.
SIAM Journal on Discrete Mathematics, Volume 4, Pages 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, Volume 54, Article 11, ACM, 2007.

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

References (cont.)

C. Nagarajan and D.P. Williamson.
Offline and Online Facility Leasing.
Integer Programming and Combinatorial Optimization, Volume 5035 in LNCS, Pages 303–315, Springer-Verlag, 2008.

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, Volume 76, Pages 709–726, Elsevier Inc., 2010.

Acknowledgements

Thank you!

Questions?

Acknowledgements

Thank you!

Questions?