Algorithmic Modeling and Complexity **Fall 2003**

Lecture 7: 23 September

Lecturer: J. van Leeuwen *Scribe: M. van Es*

7.1 Overview

Facility location problems are concerned with optimally locating facilities like warehouses, distribution centres, newspaper stands, schools, hospitals and so on, in such a way that (groups of) customers are served best at the lowest possible cost of building and operating the facilities. In this lecture we consider a typical case in networks where the 'distances' between customers and facilities satisfy a metric-symmetric property, model it as an Integer Linear Program (ILP), and attempt to solve it by a fast primal-dual type of approximation algorithm.

7.2 Modeling the Facility Location problem

A typical facility location problem is comprised of the following components:

- **locations:** places where facilities could be built and opened. Let f_i be the *cost* for doing so at location i.
- **customers/clients:** for each customer j and location (i.e. possible facility at) i, there is a *distance/service cost* c_{ij}: the cost if customer j would use service from the facility at location i.
- **requirements, wishes, and constraints:** for example, it may be required that each customer is assigned to exactly one location. A typical constraint may be that only a limited number of facilities can be built and opened.
- **goal:** to find a set of locations and an assignment of customers to them such that opening facilities at these locations and assigning customers to them leads to the least possible 'total cost'.

Definition 7.1 *The total cost of a feasible solution to the facility location problem is:*

$$\sum_{i \text{ opened}} f_i \quad + \sum_{i,j \text{ with } j \text{ assigned the opened } i} c_{ij}$$

Typically a further constraint could be imposed on the *capacity* of every facility, i.e. the maximum number of customers it can service. This constraint is not used in our case. The problem we study is therefore known as the *uncapacitated facility location problem* (UFL).

Exercise. Solve the UFL problem in case $f_i = 0$ for all facilities i. (Hint: open a facility at every location i.)

In the remainder we will assume that all $f_i \geq 0$ and $c_{ij} \geq 0$ and and that these costs are all *integer*. We will also use the terms location and facility interchangeably.

7.2.1 Representing the UFL problem

When studying the uncapacitated facility location problem, the locations can be imagined as a 'row' of nodes F (running index i) and the customers as a 'row' of nodes C (running index j) below it.

A facility or location i is labeled with its opening cost f_i. The possible assignment of customer j to facility i is described by an edge (i, j) with label c_{ij}, the already mentioned service cost.

Lemma 7.2 *The Set Cover problem can be 'reduced' to the UFL problem.*

Proof: Remember the Set Cover problem. It consists of a universe $U = \{x_1, \ldots, x_n\}$, a collection of subsets $S = \{S_1, \ldots S_m\}$ and cost value for each subset $c(S_i) = c_i$. The goal is to minimize $\sum_i c_i$.

Now design an instance of the UFL problem as follows. In the UFL problem, the customers correlate with the elements of the set U and the facilities with the subsets in the collection S. The opening cost of facility i $(1 \leq i \leq m)$ is set to $f_i = c_i$.

The question remains what to take for c_{ij} in this UFL problem instance. Take c_{ij} to be 1 if $j \in S_i$ and ∞ otherwise.

We claim that any cost-q solution of the Set Cover instance corresponds to a cost-$q+n$ solution of the constructed UFL problem instance and vice versa (for any finite q).

($\Rightarrow$) Consider any solution $S_{i_1}, \ldots, S_{i_k}$ of the Set Cover problem of cost q. In the UFL instance, open facilities $i_1, \ldots, i_k$ and assign every customer j to an opened facility that 'covers' it. (If a customer j is covered by more than one subset/facility, it has to choose one of them to connect to.) The result is a feasible solution of cost q (for the opened facilities) + n (the total cost for connecting the customers).

($\Leftarrow$) Conversely, consider any feasible solution to the UFL instance with (finite) cost $q + n$. The cost of connecting the customers is n, so the cost of the opened facilities $i_1, \ldots, i_k$ in the solution is q. It is easily verified that $S_{i_1}, \ldots, S_{i_k}$ must form feasible solution to the Set Cover instance, with cost q.

It follows that the optimum solution to the Set Cover problem equals the optimum solution of the UFL problem to which it was reduced. ∎

In general the UFL problem is computationally hard. Fortunately it can be modelled in the framework of LP.

7.3 Modeling the UFL problem by a 0-1 Linear Programming model

We now show how the UFL problem can be modelled as a 0-1 LP, we will consider the relaxed problem, take is dual and interpret the complementary slackness conditions.

Take the following indicator variables:

> y_i: indicator variable expressing whether a facility (at location) i is opened or not.
>
> x_{ij}: indicator variable expressing whether facility i will service customer j or not, i.e. whether j is serviced by i or not.

The model for minimizing the total cost of a feasible solution to a given UFL problem is due to Baliski (1966). The 0-1 LP reads as follows:

UFL$_{01}$

min $\quad \sum_{i,j} c_{ij} x_{ij} + \sum_i f_i y_i$

subject to:

1. $\sum_i x_{ij} \geq 1$ for every j (all customers must be serviced).
2. $y_i - x_{ij} \geq 0$ for all i, j (if j is connected to i, i.e. $x_{ij} = 1$ then i must be open).
3. $x_{ij} \in \{0, 1\}$.
4. $y_i \in \{0, 1\}$.

We will especially study the *relaxed* UFL$_{01}$ problem, which we will simply call 'UFL'. In the relaxed UFL the last two constraints are converted to:

$3'$. $x_{ij} \geq 0$.

$4'$. $y_i \geq 0$.

We will not consider solving the relaxed problem directly but move towards a primal-dual approach.

7.3.1 The dual UFL model and the complementary slackness conditions

The following dual variables are created:

> α_j : belonging to the constraint $\sum_i x_{ij} \geq 1$ for every j.
>
> β_{ij} : belonging to the constraint $y_i - x_{ij} \geq 0$ for all i, j.

We will see in a moment that the α_j and β_{ij} have a nice intuitive interpretation.

The dual of the UFL model is designed in the standard way:

UFL^d

max $\sum_j \alpha_j$.

subject to:

1. $\alpha_j - \beta_{ij} \leq c_{ij}$ for all i, j (dual constraint related to primal variable x_{ij}).
2. $\sum_j \beta_{ij} \leq f_i$ for all i (dual constraint related to primal variable y_i).
3. $\alpha_j \geq 0$.
4. $\beta_{ij} \geq 0$.

Looking at the dual one gets the idea for the following interpretation of the dual variables.

- α_j is the total cost paid by customer j.
- β_{ij} is the contribution of customer j to the cost of opening facility i (to which it will presumable be assigned).

Constraint 1 expresses e.g. that the amount of money customer j has left after contributing to the opening costs of facility j (i.e. $\alpha_j - \beta_{ij}$) may be used for paying towards the service costs c_{ij}. We may only hope that in an *optimum* solution, the total payments of the various customers are sufficient to cover the costs! The complementary slackness conditions express that we may indeed hope this to be the case:

Complementary slackness conditions

1. $x_{ij} > 0 \rightarrow \alpha_j - \beta_{ij} = c_{ij}$ ('if j is connected to i, then j has paid β_{ij} for the opening and has precisely c_{ij} left for the servicing costs').

2. $y_i > 0 \rightarrow \sum_j \beta_{ij} = f_i$ ('if i is opened, then all customers assigned to i have paid for opening it').

1'. $\alpha_j > 0 \rightarrow \sum_i x_{ij} = 1$ ('if customer j pays an amount α_j, then it is covered by the facilities'). then it is covered by a facility.)

2'. $\beta_{ij} > 0 \rightarrow y_i = x_{ij}$ ('if customer j contributes to the cost of opening a facility then this facility is opened and j is connected to it').

The only difficulty is that the y_i and x_{ij} are not necessarily 0-1 valued in an optimal solution to UFL and therefore the α_j and β_{ij} cannot automatically be assumed to have the desired interpretation either.

7.3.2 An ideal primal-dual argument

Jain and Vazirani (2001) suggested the possibility to solve the UFL^d by trying to satisfying the complementary slackness conditions i.e. by using the primal-dual argument. The idea is, that if we have a feasible solution in the dual model and can 'move' this towards the optimum while maintaining values for x_{ij} and y_i that eventually become feasible as well, then the final feasible solution to the primal that is found this way may be a good approximate solution to the original UFL$_{01}$ problem.

Suppose we manage to find solutions to the dual and primal problem such that:

- $x_{ij}, y_i \in \{0,1\}$ are primal feasible. Let the assignment ϕ be such that $\phi(j) = i$ if and only if $x_{ij} = 1$.
- α_j and β_{ij} are dual feasible (we will later see that we can even take them to be *integral* but this is not needed here).
- the dual complementary slackness conditions hold, i.e.

  1$'$. $\alpha_j > 0 \rightarrow (\sum_i x_{ij} = 1 \rightarrow$ by 1. there is precisely one x_{ij} equal to1 thus $\phi(j) = i$ hence $) \rightarrow x_{\phi(j),j} = 1$.

  2$'$. $\beta_{ij} > 0 \rightarrow y_i = x_{ij}$ (thus if $x_{ij} = 1$ then $y_i = 1$, meaning that facility i is open).

- the *weakened* primal slackness conditions hold,

  1. $x_{ij} > 0 \rightarrow \frac{1}{3} c_{ij} \leq \alpha_j - \beta_{ij} \leq c_{ij}$ (with $i = \phi(j)$).
  2. $y_i > 0 \rightarrow \frac{1}{3} f_i \leq \sum_{\phi(j) = i} \beta_{ij} \leq f_i$.

Proposition 7.3 *If the assumption holds, the primal solution yields an approximation of the UFL$_{01}$ problem within a factor* 3 *from optimum.*

Proof: If the assumption holds, then consider the cost of the solution y_i, x_{ij}:

$\sum c_{ij} x_{ij} + \sum f_i y_i$

$\leq \sum c_{ij} x_{ij} + \sum 3\beta_{ij} x_{ij}$
(looking only at terms $f_i y_i$ with $y_i = 1$ we can estimate f_i by $3 \cdot \sum_{\phi(j) = i} \beta_{ij}$. Replacing the factor y_i by x_{ij} is harmless because whenever $\beta_{ij} \neq 0$, we have $y_i = x_{ij}$ by assumption. Extending the summation over all i, j is fine too as it only increases our sum.)

$= \sum (c_{ij} + 3\beta_{ij}) x_{ij}$
(only the terms with $x_{ij} \neq 0$ count, thus use the weak primary slackness constraint)

$\leq \sum_j 3\alpha_j$

$\leq 3\cdot$ optimal value of the dual $= 3\cdot$ optimal value of the primal

$\leq 3\cdot$ optimal of the original UFL$_{01}$ model.

This proves the claim. ■

Jain and Vazirani claim that if we assume that the UFL problem can be embedded in a metric-symmetric network situation (i.e. for the costs c_{ij}), then we can achieve the assumed properties, even in a strong way.

7.4 A primal-dual approach to finding a feasible solution

Assume that the costs c_{ij} satisfy a metric-symmetric property (see Lemma 7.10). Also recall that the f_i and c_{ij} are assumed to be *integral*.

The aim is more or less clear. We want to start with a feasible dual solution and build towards a *large* solution satisfying the complementary slackness conditions. We will do this in terms of the α_j and β_{ij} first: it means that we will let a sufficient number, but not too many customers 'pay' for the opening and service costs through their 'dual' variables. Setting 0-1 values for the y_i and x_{ij} will be done at the end. Hopefully the result is a solution that is 'close to optimal'.

Starting with a dual feasible solution is easy. We take $\alpha_j = \beta_{ij} = 0$. We now grow a 'large' dual feasible solution by incrementing the α_j by 1 repeatedly, and if necessary the β_{ij} too, while maintaining dual feasibility, i.e.

$$\alpha_j - \beta_{ij} \leq c_{ij} \text{ and } \sum \beta_{ij} \leq f_i.$$

Raise α_j until $\alpha_j - \beta_{ij} = c_{ij}$ holds, then raise α_j and β_{ij} by 1 *together* (to keep their difference at c_{ij}) until $\sum \beta_{ij} = f_i$ holds. How far we go with this depends on several possible *events*.

As soon as some of the β_{ij}-values are beginning to rise, we may get to a point where 'equality' arises in some or more of the constraints $\sum \beta_{ij} \leq f_i$, meaning that facility i gets fully paid and 'can be built if we want to' (thus not necessarily opened).

Definition 7.4 *Facility i is* fully paid for *if* $\sum_j \beta_{ij} = f_i$.

Definition 7.5 *Customer j is paying for facility i if $\beta_{ij} > 0$.*

Clearly, if a facility i is fully paid for, then no customer j should raise his β_{ij} further. It follows that *if* j had already raised its α_j so far that it was forced to raise its β_{ij}'s along with it, then it should not raise its α_j any further either! Otherwise it will continue to raise its α_j until it satisfies some $\alpha_j \geq c_{ij}$ for the first time.

Definition 7.6 *Customer j has* reached *facility i if $\alpha_j \geq c_{ij}$. If facility i is the 'first' among the facilities reached by j to become fully paid for, then j is said to become* linked *to i and i is* the linking facility *for j.*

(The 'status' of a customer depends on the progress of the algorithm. In the definition it is to be understood that once a customer j has reached a facility i, it will have reached i from now on. Typically j's dual variables are no longer raised once it has become linked to a facility.)

Note that facilities that are fully paid for are not necessarily all going to be opened. Neither are customers that get *linked* to a facility i automatically going to be assigned to i in the end. We will partition the customers in groups later.

We describe the algorithm below. It uses three checks:

Check-reached(j)

only called when j is not *linked and its α_j has just been incremented*

check whether j has 'reached' a new i

going through all i **do**

  if $\alpha_j \geq c_{ij}$ **then** declare i reached by j

end

Check-fullypaid(i)

only called when i is not *yet fully paid and some j just raised its α_j and β_{ij}*

if $\sum_j \beta_{ij} = f_i$ **then** declare i fully paid

end

Check-linked(j)

only called when j is not *linked and some i has just become fully paid*

going through all i that are fully paid for **do**

  if j has reached i and is not yet linked **then** declare j linked to i (with i the linking facility for j)

  Note that this links j arbitrarily to the first fully paid i that it has reached. Note also that several j may be linked to the same i. Once a j gets linked it will not be linked again, by virtue of the if-test

end

The three checks could be described more efficiently e.g. the various sums checked need not be evaluated each time from scratch but can be maintained incrementally. We ignore this detail now. With the three checks the algorithm 'JV' becomes as follows.

Algorithm JV

Loop so all j eventually become linked.
Use a loop counter t as a 'clock'

$t := 0$

for all unlinked j **do Check-reached**(j)

for all i not fully paid for **do Check-fullypaid**(i)

for all unlinked j **do Check-linked**(j)

while there are unlinked customers left **do**

begin

> *Now raise the contribution of the unlinked customers. While raising the contributions, customers may reach more facilities, facilities may become fully paid for and, subsequently, customers may become linked in the process*
>
> $t := t + 1$
>
> **for** every unlinked j **do**
>
> **begin**
>
>> *Raise j's contribution by 1 unit*
>> $\alpha_j := \alpha_j + 1$
>> *and check whether any β_{ij} must be increased too*
>> **for** every facility i **do**
>> **begin**
>>> **if** j did not reach i (i.e. before now) **then**
>>>> *the increase of α_j was all*
>>>
>>> **else if** j has reached i **then**
>>>> *because j is unlinked, i cannot be fully paid for yet*
>>>> $\beta_{ij} := \beta_{ij} + 1$
>>
>> **end**
>> *do the final bookkeeping for j*
>> **Check-reached**(j)
>> **for all** i not fully paid for **do Check-fullypaid**(i)
>> *any i that became fully paid for may now serve as linking facility*
>> **for all** unlinked j **do Check-linked**(j)
>
> **end**

end

Exercise. Show that algorithm JV must terminate after finitely many steps i.e. after incrementing the the dual variables finitely many times.

7.4.1 Properties of algorithm JV

We will now argue that algorithm JV does the job we want. We first show some auxiliary properties. This part of our exposition elaborates on the approach in [2].

Definition 7.7 *For every facility i that is fully paid for, let t_i be the t-value during the algorithm at which facility i became fully paid for.*

Lemma 7.8 *a. If customer j has reached i, then $\alpha_j \geq c_{ij}$.*

b. If $\beta_{ij} > 0$ (i.e. j 'pays' for facility i) and facility i is fully paid for, then $\alpha_j \leq t_i$.

c. If i is the linking facility for j, then $t_i \leq \alpha_j$.

Proof:

(a.) This is immediate from the definition of **Check-reached**.

(b.) $\beta_{ij} > 0$ means that j reaches i before i is fully paid for (otherwise j was linked immediately). α_j is thus incremented further in the rounds of algorithm JV after j reaches i, until j gets linked. Clearly j gets linked the latest when i becomes fully paid for.

(c.) Let i be the linking facility for j. There are two possibilities:

Case 1. $\beta_{ij} > 0$. In this case we have $\alpha_j = t_i$, because the incrementing of α_j went along with the incrementing of t exactly until the moment of linking.

Case 2. $\beta_{ij} = 0$. If j reached i (at time α_j), then this must be after i got fully paid for. Thus $t_i \leq \alpha_j$. ■

We now move towards defining a suitable primal solution, especially the y_i that should become 1 ('which facilities will be opened'). We introduce:

Definition 7.9 *The domain of i is the set $D_i = \{j|\beta_{ij} > 0\}$, i.e. the set of all customers that pay for i.*

What can we say about the domains D_i. The customers that pay for facilities are divided over the domains but may belong to several domains simultaneously ('pay for several facilities'). And there may be a bunch of customers that do not pay for facilities at all, i.e. have all their $\beta_{ij} = 0$.

The following lemma is crucial and will use the symmetric and metric property of the costs, viz. the *triangle inequality* for the c_{ij}'s.

Lemma 7.10 *Let i and i' be facilities that are fully paid for and suppose $D_i \cap D_{i'} \neq \emptyset$. Let i' be the linking facility for j. Then*

$$\frac{1}{3}c_{ij} \leq \alpha_j - \beta_{ij} \leq c_{ij}.$$

Proof:

Consider customer j and distinguish two cases.

Case 1: j has reached i. Then by the algorithm we have $\alpha_j - \beta_{ij} = c_{ij}$ so this is ok.

Case 2: j did not reach i. This means that necessarily $\beta_{ij} = 0$ and $\alpha_j \leq c_{ij}$, which gives the right side of the inequality.

Let $j' \in D_i \cap D_{i'}$. Now j' must have reached both i and i', thus by the previous lemma: $c_{ij'} \leq \alpha_{j'}$ and $c_{i'j'} \leq \alpha_{j'}$. Because j' is paying for both facilities we have $\beta_{ij'} > 0$ and $\beta_{i'j'} > 0$ and hence, by the previous lemma: $\alpha_{j'} \leq t_{i'}$ and $\alpha_{j'} \leq t_i$. Furthermore, i' was the linking facility for j. This means that $c_{i'j} \leq \alpha_j$ and that by the lemma: $t_{i'} \leq \alpha_j$. Combining all this gives, using symmetry and the triangle inequality:

$$c_{ij} \leq c_{ij'} + c_{i'j'} + c_{i'j} \leq \alpha_{j'} + \alpha_{j'} + \alpha_j \leq 3 \cdot \alpha_j$$

where we use that $\alpha_{j'} \leq \min\{t_{i'}, t_i\} \leq \alpha_j$. Thus we have in this case: $\frac{1}{3} c_{ij} \leq \alpha_j - \beta_{ij}$ as desired. (Note that $\beta_{ij} = 0$ here.) ∎

7.4.2 Opening facilities and assigning customers

We now have all ingredients for determining a 'good' primal solution. Recall that our aim is to satisfy the conditions from Section 7.3.2, especially the dual and *weakened* primal complementary slackness conditions.

Let FP be the set of fully paid facilities. Which ones are we going to open and how do we assign customers?

Definition 7.11 *Let I be a* maximal *set of facilities $i_1, \ldots, i_r \in FP$ such that their D_i's are mutually disjoint.*

Open the facilities $\in I$, i.e. set $y_i = 1$ for $i \in I$. Assign customers j to open facilities $\phi(j)$ as follows:

> if $j \in D_i$ for an $i \in I$ then assign j to i: $\phi(j) = i$. We declare j 'directly connected'.
>
> if j does *not* belong to any D_i with $i \in I$, then
>
> > let i' be the linking facility for j (thus $i' \in FP$)
> >
> > if $i' \in I$ (then necessarily $\beta_{i'j} = 0$ because $j \notin I$ and) we assign j to i' again: $\phi(j) = i'$. We declare j 'directly connected'.
> >
> > if $i' \notin I$ then let $i \in I$ be such that $D_i \cap D_{i'} \neq \emptyset$. (Because I is maximal, such an i must exist.) Now assign j to i: $\phi(j) = i$. We declare j 'indirectly connected'.

Observe that all customers j are assigned and that they are all assigned to an opened facility! Set $x_{ij} = 1$ if and only if $\phi(j) = i$.

Theorem 7.12 (Jain and Vazirani, 2001) *The y_i, x_{ij} form a feasible solution to the UFL_{01} problem that is within a factor 3 from optimum.*

Proof: The solution y_i, x_{ij} is a 0-1 solution that is clearly feasible by construction. The α_j, β_{ij} in turn are a dual feasible solution, again by construction. We proceed by showing that the primal and dual feasible solutions we have, satisfy the slackness conditions $1', 2', 1$ and 2 as given in Section 7.3.2.

Ad 1'. This is trivial, as ϕ is a valid assignment for *all* customers j.

Ad 2'. Let $\beta_{ij} > 0$. Consider customer j. If $i \in I$ then i is open and j is assigned to i: $y_i = x_{ij} = 1$. If $i \notin I$ then i is not open and j cannot be assigned to it: $y_i = x_{ij} = 0$.

Ad 1. Let $x_{ij} > 0$, hence $x_{ij} = 1$: j is assigned to opened facility $i \in I$. If j is 'directly connected' to i, then necessarily $\alpha_j - \beta_{ij} = c_{ij}$ thus the inequality holds. If j is indirectly connected to i, then Lemma 7.10 applies, proving the inequality again!

Ad 2. Let $y_i > 0$, hence $y_i = 1$: thus i is open and $i \in I$. By the assignment method, all customers j with $\beta_{ij} > 0$ are assigned to i, i.e. have $\phi(j) = i$. Because $i \in FP$, it is fully paid for and we have

$$\sum \ \phi(j) = i \quad \beta_{ij} = \sum \beta_{ij} = f_i$$

which is even stronger than required.

The theorem now follows from Proposition 7.3. ■

7.4.3 Further remarks

Jain and Vazirani [3] show that algorithm JV can be implemented so as to run in $O(m \log m)$ time, where $m = |F| \cdot |C|$.

The *uncapacitated facility location problem* has enjoyed considerable interest in the last few years, resulting in ever better polynomial time approximation algorithms. We list some of them below, with the performance ratio they achieve.

year	author(s)	performance ratio
1997	Shmoys, Tardos and Aardal	3.16
1999	Jain and Vazirani	3
1999	Guha and Khuller	2.47
1999	Charikar and Guha	1.72
2001	Jain, Mahdian and Saberi	1.61
2002	Mahdian, Ye, and Zhang	1.52

References

[1] M.L. Baliski. On finding integer solutions to linear programs. In: *Proc. IBM Scientific Computing Symp. on Combinatorial Problems*, 1966, pp. 225-248.

[2] A. Bumb. *Approximation algorithms for facility location problems*, PhD. Thesis, Twente University Press, Enschede, 2002.

[3] K. Jain, V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *Journal of the ACM* 48:2 (2001) 274-296.