Network Design and Optimization course

Lecture 8

Alberto Ceselli
alberto.ceselli@unimi.it

Dipartimento di Tecnologie dell'Informazione
Università degli Studi di Milano

November 21, 2011

The problem

Given

- a set of nodes,
- a set of links connecting them,
- a set of **service requests**, **one for each node** of the network,
- a set of **devices**, able to provide service, to be installed in the network,

I want to

- decide where to place the service provider devices,
- decide how to satisfy service requests,
- maximizing the quality of service (e.g. minimizing delay time)

Locating devices on a network

It is not a single problem, but a *class* of problems, known as **Facility Location**.

Several modeling issues arise:

Locating devices on a network

It is not a single problem, but a *class* of problems, known as **Facility Location**.

Several modeling issues arise:

- is the capacity of a service provider finite?

Locating devices on a network

It is not a single problem, but a *class* of problems, known as **Facility Location**.
Several modeling issues arise:

- is the capacity of a service provider finite?
- is it possible to answer to the same service request using (fracionally) different providers?

Locating devices on a network

It is not a single problem, but a *class* of problems, known as **Facility Location**.
Several modeling issues arise:

- is the capacity of a service provider finite?
- is it possible to answer to the same service request using (fracionally) different providers?
- is there a limit on the *number* of service requests that can be served by the same provider (e.g. devices with limited number of ports)?

Locating devices on a network

It is not a single problem, but a *class* of problems, known as **Facility Location**.
Several modeling issues arise:

- is the capacity of a service provider finite?
- is it possible to answer to the same service request using (fracionally) different providers?
- is there a limit on the *number* of service requests that can be served by the same provider (e.g. devices with limited number of ports)?
- is the problem single or multi commodity?

Locating devices on a network

It is not a single problem, but a *class* of problems, known as **Facility Location**.

Several modeling issues arise:

- is the capacity of a service provider finite?
- is it possible to answer to the same service request using (fracionally) different providers?
- is there a limit on the *number* of service requests that can be served by the same provider (e.g. devices with limited number of ports)?
- is the problem single or multi commodity?
- how to measure the overall quality of service?

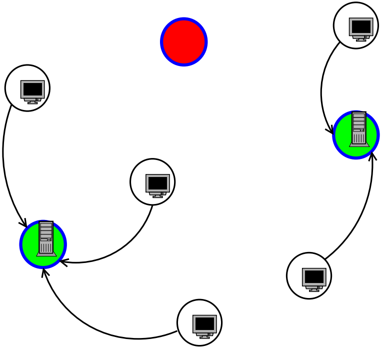
Assumptions

Let us begin with a basic location problem:

1. providers with very large resources (capacity and number of ports are not an issue),
2. possibility of splitting service requests,
3. single commodity,
4. minimize the average *connection cost*.

In this case we talk about a **Uncapacitated Facility Location Problem**.

Visually:



Problem features:

Given:

- A graph $G(V, E)$ (telecomunication network: $V =$ sites, $E =$ links).

Problem features:

Given:

- A graph $G(V, E)$ (telecomunication network: $V =$ sites, $E =$ links).
- A subset I of vertices of the graph, which correspond to sites in which servers can be installed.
- A subset J of vertices of the graph, in which terminals are placed.

Problem features:

Given:

- A graph $G(V, E)$ (telecomunication network: $V =$ sites, $E =$ links).
- A subset I of vertices of the graph, which correspond to sites in which servers can be installed.
- A subset J of vertices of the graph, in which terminals are placed.
- Installing a server in each site $i \in I$ has a cost f_i.
- Connecting a terminal in site $j \in J$ to a server in $i \in I$ has a cost c_{ij}.

Problem features:

Given:

- A graph $G(V, E)$ (telecomunication network: $V =$ sites, $E =$ links).
- A subset I of vertices of the graph, which correspond to sites in which servers can be installed.
- A subset J of vertices of the graph, in which terminals are placed.

- Installing a server in each site $i \in I$ has a cost f_i.
- Connecting a terminal in site $j \in J$ to a server in $i \in I$ has a cost c_{ij}.

- Choose if and where to intall the servers (binary variables y_i) and how to connect terminals to servers (variables x_{ij})...

Problem features:

Given:

- A graph $G(V, E)$ (telecomunication network: $V =$ sites, $E =$ links).
- A subset I of vertices of the graph, which correspond to sites in which servers can be installed.
- A subset J of vertices of the graph, in which terminals are placed.

- Installing a server in each site $i \in I$ has a cost f_i.
- Connecting a terminal in site $j \in J$ to a server in $i \in I$ has a cost c_{ij}.

- Choose if and where to intall the servers (binary variables y_i) and how to connect terminals to servers (variables x_{ij})...
- ...in such a way that each terminal is connected to a server.

Uncapacitated Facility Location Problem (UFLP)

$$
\begin{aligned}
\min \quad & \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} & \\
\text{s.t.} \quad & \sum_{i \in I} x_{ij} = 1 & \forall j \in J \\
& x_{ij} \leq y_i & \forall i \in I, \forall j \in J \\
& x_{ij} \geq 0 & \forall i \in I, \forall j \in J \\
& y_i \in \{0, 1\} & \forall i \in I
\end{aligned}
$$

N.B. Without additional conditions, variables x_{ij} take integer values.

Solving UFLP

- The problem is “difficult” (NP-Hard);

Solving UFLP

- The problem is “difficult” (NP-Hard);
- we try to use the *Lagrangean Relaxation* technique in order to get lower (dual) bounds to the problem;

Solving UFLP

- The problem is "difficult" (NP-Hard);
- we try to use the *Lagrangean Relaxation* technique in order to get lower (dual) bounds to the problem;
- we try to *repair* a (generally infeasible) solution of the relaxation to obtain a feasible solution to the problem;

Solving UFLP

- The problem is "difficult" (NP-Hard);
- we try to use the *Lagrangean Relaxation* technique in order to get lower (dual) bounds to the problem;
- we try to *repair* a (generally infeasible) solution of the relaxation to obtain a feasible solution to the problem;
- this solution provides an upper (primal) bound: might not be optimal, but in general good enough;

Solving UFLP

- The problem is "difficult" (NP-Hard);
- we try to use the *Lagrangean Relaxation* technique in order to get lower (dual) bounds to the problem;
- we try to *repair* a (generally infeasible) solution of the relaxation to obtain a feasible solution to the problem;
- this solution provides an upper (primal) bound: might not be optimal, but in general good enough;
- the difference between upper and lower bound gives a *quality estimation* of our solution;

Solving UFLP

- The problem is "difficult" (NP-Hard);
- we try to use the *Lagrangean Relaxation* technique in order to get lower (dual) bounds to the problem;
- we try to *repair* a (generally infeasible) solution of the relaxation to obtain a feasible solution to the problem;
- this solution provides an upper (primal) bound: might not be optimal, but in general good enough;
- the difference between upper and lower bound gives a *quality estimation* of our solution;
- if the quality is not enough, we resort to branch-and-bound (enumeration).

UFLP: Lagrangean Relaxation

$$\min z = \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_{i \in I} x_{ij} = 1 \qquad \forall j \in J$$

$$y_i \geq x_{ij} \qquad \forall i \in I, \forall j \in J$$

$$x_{ij} \geq 0 \qquad \forall i \in I, \forall j \in J$$

$$y_i \in \{0, 1\} \qquad \forall i \in I$$

UFLP: Lagrangean Relaxation

$$\min z = \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

$$\begin{aligned}
\text{s.t.} \quad & \sum_{i \in I} x_{ij} - 1 \geq 0 && \forall j \in J \\
& y_i - x_{ij} \geq 0 && \forall i \in I, \forall j \in J \\
& x_{ij} \geq 0 && \forall i \in I, \forall j \in J \\
& y_i \in \{0,1\} && \forall i \in I
\end{aligned}$$

UFLP: Lagrangean Relaxation

$$\min z = \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} - \sum_{j \in J} \pi_j \Big(\sum_{i \in I} x_{ij} - 1\Big) - \sum_{i \in I} \sum_{j \in J} \eta_{ij} (y_i - x_{ij})$$

$$\text{s.t.} \quad \sum_{i \in I} x_{ij} - 1 \geq 0 \quad \forall j \in J \ (\underline{\pi})$$

$$y_i - x_{ij} \geq 0 \quad \forall i \in I, \forall j \in J \ (\underline{\eta})$$

$$x_{ij} \geq 0 \quad \forall i \in I, \forall j \in J$$

$$y_i \in \{0, 1\} \quad \forall i \in I$$

UFLP: Lagrangean Relaxation

$$w(\underline{x}, \underline{y}, \underline{\pi}, \underline{\eta}) = \min \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} - \sum_{j \in J} \pi_j (\sum_{i \in I} x_{ij} - 1) - \sum_{i \in I} \sum_{j \in J} \eta_{ij} (y_i - x_{ij})$$

s.t.

$$x_{ij} \geq 0 \quad \forall i \in I, \forall j \in J$$

$$y_i \in \{0, 1\} \quad \forall i \in I$$

UFLP: Lagrangean Relaxation

$$w(\underline{x}, \underline{y}, \underline{\pi}, \underline{\eta}) = \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \\ - \sum_{j \in J} \pi_j (\sum_{i \in I} x_{ij} - 1) \quad - \sum_{i \in I} \sum_{j \in J} \eta_{ij} (y_i - x_{ij})$$

$$L(\underline{\pi}, \underline{\eta}) = \min_{\underline{x}, \underline{y}} \; w(\underline{x}, \underline{y}, \underline{\pi}, \underline{\eta})$$

$$\text{s.t. } 0 \leq x_{ij} (\leq 1) \qquad \forall i \in I, \forall j \in J$$

$$y_i \in \{0, 1\} \qquad \forall i \in I$$

$$u_{ij} \geq 0 \qquad \forall i \in I, \forall j \in J$$

$w(\underline{x}, \underline{y}, \underline{\pi}, \underline{\eta})$ is called the *Lagrangean Function* and $\max_{\underline{\pi}, \underline{\eta}} L(\underline{\pi}, \underline{\eta})$ is called the *Lagrangean Dual Problem*.

Solving the Relaxation

- For *any* choice of $\underline{\pi}$ and $\underline{\eta}$, $L(\underline{\pi}, \underline{\eta})$ gives a valid lower (dual) bound bound to the value of z^*.

Solving the Relaxation

- For *any* choice of $\underline{\pi}$ and $\underline{\eta}$, $L(\underline{\pi}, \underline{\eta})$ gives a valid lower (dual) bound bound to the value of z^*.
- In order to obtain the *tightest* lower bound we had to solve the Lagrangean Dual Problem to optimality

Solving the Relaxation

- For *any* choice of $\underline{\pi}$ and $\underline{\eta}$, $L(\underline{\pi}, \underline{\eta})$ gives a valid lower (dual) bound bound to the value of z^*.
- In order to obtain the *tightest* lower bound we had to solve the Lagrangean Dual Problem to optimality
- Possible from a theoretical point of view, very hard from a computational point of view.

Solving the Relaxation

- For *any* choice of $\underline{\pi}$ and $\underline{\eta}$, $L(\underline{\pi}, \underline{\eta})$ gives a valid lower (dual) bound bound to the value of z^*.
- In order to obtain the *tightest* lower bound we had to solve the Lagrangean Dual Problem to optimality
- Possible from a theoretical point of view, very hard from a computational point of view.
- We resort to *approximate* solutions for the Lagrangean Dual Problem.

Solving the Relaxation

- Lagrangean Relaxation was first devised to solve *nonlinear (continuous)* problems.

Solving the Relaxation

- Lagrangean Relaxation was first devised to solve *nonlinear (continuous)* problems.
- There are *many* iterative algorithms (update variables and multipliers until convergence).

Solving the Relaxation

- Lagrangean Relaxation was first devised to solve *nonlinear (continuous)* problems.
- There are *many* iterative algorithms (update variables and multipliers until convergence).
- The most simple choice is the *gradient* algorithm. Iteratively:
    - fix the multipliers and find a (local) optimum with respect to the remaining variables,
    - compute the gradient of the Lagrangean Dual Function with respect to the multipliers,
    - update the multipliers according to these gradients

  until the gradients are 0 (or early termination criteria).

Solving the Relaxation

- Lagrangean Relaxation was first devised to solve *nonlinear (continuous)* problems.
- There are *many* iterative algorithms (update variables and multipliers until convergence).
- The most simple choice is the *gradient* algorithm. Iteratively:
  - fix the multipliers and find a (local) optimum with respect to the remaining variables,
  - compute the gradient of the Lagrangean Dual Function with respect to the multipliers,
  - update the multipliers according to these gradients

  until the gradients are 0 (or early termination criteria).
- Our Lagrangean Dual Function is *nondifferentiable* (piecewise linear), hence we use *subgradients* instead of gradients.

Subgradient algorithm for UFLP (1)

- Choose an initial value for the multipliers $\underline{\pi}^0$ e $\underline{\eta}^0$.

Subgradient algorithm for UFLP (1)

- Choose an initial value for the multipliers $\underline{\pi}^0$ e $\underline{\eta}^0$.
- Iteratively $(k = 1 \ldots)$
  - *Solve the Lagrangean Subproblem* in $\underline{x}$ e $\underline{y}$, getting a solution $(\underline{x}^k, \underline{y}^k)$ of value ω^k.
  - Compute the subgradients:
    - $\nabla h_j^k = 1 - \sum_{i \in I} x_{ij}^k$
    - $\nabla g_{ij}^k = x_{ij}^k - y_i^k$
  - Choose a step length α^k.
  - Update the multipliers:
    - $\pi_j^{k+1} = \pi_j^k - \alpha^k \cdot \nabla h_j^k$
    - $\eta_{ij}^{k+1} = \eta_{ij}^k - \alpha^k \cdot \nabla g_{ij}^k$

Subgradient algorithm for UFLP (2)

A popular choice is

- to set the step length α^k according to the rule:

$$\alpha^k = \rho^k \cdot \frac{\text{UB} - \omega^k}{\sum_{i \in I, j \in J} (\nabla g_{ij}^k)^2 + \sum_{j \in J} (\nabla h_j^k)^2}$$

where UB is an upper bound to the optimal value of $w(\underline{x}, \underline{y}, \underline{\pi}, \underline{\eta})$.

Subgradient algorithm for UFLP (2)

A popular choice is

- to set the step length α^k according to the rule:

$$\alpha^k = \rho^k \cdot \frac{\mathrm{UB} - \omega^k}{\sum_{i \in I, j \in J} (\nabla g_{ij}^k)^2 + \sum_{j \in J} (\nabla h_j^k)^2}$$

  where UB is an upper bound to the optimal value of $w(\underline{x}, \underline{y}, \underline{\pi}, \underline{\eta})$.

- to (maybe) reduce the parameter ρ^k at some iteration according to the rule:

$$\rho^{k+1} = 0.5 \cdot \rho^k$$

Subgradient algorithm for UFLP (2)

A popular choice is

- to set the step length α^k according to the rule:

$$\alpha^k = \rho^k \cdot \frac{\mathrm{UB} - \omega^k}{\sum_{i \in I, j \in J} (\nabla g_{ij}^k)^2 + \sum_{j \in J} (\nabla h_j^k)^2}$$

  where UB is an upper bound to the optimal value of $w(\underline{x}, \underline{y}, \underline{\pi}, \underline{\eta})$.
- to (maybe) reduce the parameter ρ^k at some iteration according to the rule:

$$\rho^{k+1} = 0.5 \cdot \rho^k$$

- Moreover, if for some (i,j) $\eta_{ij}^{k+1} \leq 0$, in order to fulfil constraints on the sign of multipliers $\underline{\eta}$, set $\eta_{ij}^{k+1} = 0$.

Subgradient algorithm for UFLP (3)

- Iterate until
  - $\nabla g_{ij}^k = 0$ and $\nabla h_j^k = 0$ for each $i \in I, j \in J$ (optimum reached), or
  - a *maximum* number of iterations is performed, or
  - the value ω^k is *sufficiently* near to UB, or
  - the value ρ is *sufficiently* small

In the first case, the Lagrangean Dual Problem has been solved to optimality, otherwise we have no guarantee on the quality of the (dual) solution.

Solving the Lagrangean Subproblem (1)

$$\min_{\underline{x},\underline{y}} \sum_{i\in I} f_i y_i + \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} +$$

$$- \sum_{j\in J} \pi_j (\sum_{i\in I} x_{ij} - 1) - \sum_{i\in I, j\in J} \eta_{ij}(y_i - x_{ij})$$

$$\text{s.t. } 0 \leq x_{ij} \leq 1 \qquad \forall i \in I, \forall j \in J$$

$$y_i \in \{0,1\} \qquad \forall i \in I$$

$$u_{ij} \geq 0 \qquad \forall i \in I, \forall j \in J$$

Solving the Lagrangean Subproblem (2)

$$\begin{aligned}
\min_{\underline{x},\underline{y}} \quad & \sum_{i\in I}(f_i - \sum_{j\in J}\eta_{ij})y_i + \sum_{i\in I}\sum_{j\in J}(c_{ij} - \pi_j + u_{ij})x_{ij} - \sum_{j\in J}\pi_j \\
\text{s.t.} \quad & 0 \le x_{ij} \le 1 && \forall i \in I, \forall j \in J \\
& y_i \in \{0,1\} && \forall i \in I \\
& u_{ij} \ge 0 && \forall i \in I, \forall j \in J
\end{aligned}$$

UNIVERSITÀ DEGLI STUDI DI MILANO

Solving the Lagrangean Subproblem (2)

$$\begin{aligned}
\min_{\underline{x},\underline{y}} \ & \sum_{i\in I}(f_i - \sum_{j\in J}\eta_{ij})y_i + \quad \sum_{i\in I}\sum_{j\in J}(c_{ij} - \pi_j + u_{ij})x_{ij} - \sum_{j\in J}\pi_j \\
\text{s.t. } & 0 \leq x_{ij} \leq 1 & \forall i \in I, \forall j \in J \\
& y_i \in \{0,1\} & \forall i \in I \\
& u_{ij} \geq 0 & \forall i \in I, \forall j \in J
\end{aligned}$$

- Set $\tilde{f}_i = (f_i - \sum_{j\in J}\eta_{ij})$ and $\tilde{c}_{ij} = (c_{ij} - \pi_j + \eta_{ij})$.

Solving the Lagrangean Subproblem (2)

$$\begin{aligned}
\min_{\underline{x},\underline{y}} \ & \sum_{i \in I}(f_i - \sum_{j \in J} \eta_{ij})y_i + \quad \sum_{i \in I}\sum_{j \in J}(c_{ij} - \pi_j + u_{ij})x_{ij} - \sum_{j \in J} \pi_j \\
\text{s.t.} \ & 0 \leq x_{ij} \leq 1 && \forall i \in I, \forall j \in J \\
& y_i \in \{0,1\} && \forall i \in I \\
& u_{ij} \geq 0 && \forall i \in I, \forall j \in J
\end{aligned}$$

- Set $\tilde{f}_i = (f_i - \sum_{j \in J} \eta_{ij})$ and $\tilde{c}_{ij} = (c_{ij} - \pi_j + \eta_{ij})$.
- If $\tilde{f}_i \geq 0$ set $y_i = 0$, otherwise set $y_i = 1$;
- If $\tilde{c}_{ij} \geq 0$ set $x_{ij} = 0$, otherwise set $x_{ij} = 1$

Solving the Lagrangean Subproblem (3)

- Question: what is changing by relaxing constraints $y_i \in \{0, 1\}$ to $0 \leq y_i \leq 1$?

Solving the Lagrangean Subproblem (3)

- Question: what is changing by relaxing constraints $y_i \in \{0, 1\}$ to $0 \leq y_i \leq 1$?
- Answer: nothing!

Solving the Lagrangean Subproblem (3)

- Question: what is changing by relaxing constraints $y_i \in \{0, 1\}$ to $0 \leq y_i \leq 1$?
- Answer: nothing!
- The Lagrangean Subproblems has the *intergality property*: there exist optimal solutions to its continuous relaxation in which all the variables take integer values.

Solving the Lagrangean Subproblem (3)

- Question: what is changing by relaxing constraints $y_i \in \{0,1\}$ to $0 \leq y_i \leq 1$?
- Answer: nothing!
- The Lagrangean Subproblems has the *intergality property*: there exist optimal solutions to its continuous relaxation in which all the variables take integer values.
- In this case, computing a lower bound to the integer problem or computing a lower bound *to its continuous relaxation* is *the same*.

Solving the Lagrangean Subproblem (3)

- Question: what is changing by relaxing constraints $y_i \in \{0, 1\}$ to $0 \leq y_i \leq 1$?
- Answer: nothing!
- The Lagrangean Subproblems has the *intergality property*: there exist optimal solutions to its continuous relaxation in which all the variables take integer values.
- In this case, computing a lower bound to the integer problem or computing a lower bound *to its continuous relaxation* is *the same*.
- Result: the lower bound given by Lagrangean Relaxation in *not tighter* than the lower bound given by the continuous relaxation of the problem.

Solving the Lagrangean Subproblem (3)

- Question: what is changing by relaxing constraints $y_i \in \{0, 1\}$ to $0 \leq y_i \leq 1$?
- Answer: nothing!
- The Lagrangean Subproblems has the *intergality property*: there exist optimal solutions to its continuous relaxation in which all the variables take integer values.
- In this case, computing a lower bound to the integer problem or computing a lower bound *to its continuous relaxation* is *the same*.
- Result: the lower bound given by Lagrangean Relaxation in *not tighter* than the lower bound given by the continuous relaxation of the problem.
- When the problem is continuous, by solving the LD Problem we obtain *the optimum* of the original problem.

Remarks

- If the Lagrangean Subproblem has the integrality property, the corresponding lower bound is not tighter than the continuous relaxation lower bound ...

Remarks

- If the Lagrangean Subproblem has the integrality property, the corresponding lower bound is not tighter than the continuous relaxation lower bound ...
- ...but might be easier to compute!

Remarks

- If the Lagrangean Subproblem has the integrality property, the corresponding lower bound is not tighter than the continuous relaxation lower bound . . .
- . . . but might be easier to compute!
- The subgradient algorithm is generic.

Remarks

- If the Lagrangean Subproblem has the integrality property, the corresponding lower bound is not tighter than the continuous relaxation lower bound ...
- ...but might be easier to compute!
- The subgradient algorithm is generic.
- An idea is to exploit the equivalence between multipliers and dual variables in a Linear Programming problem,

Remarks

- If the Lagrangean Subproblem has the integrality property, the corresponding lower bound is not tighter than the continuous relaxation lower bound . . .
- . . . but might be easier to compute!
- The subgradient algorithm is generic.
- An idea is to exploit the equivalence between multipliers and dual variables in a Linear Programming problem,
- and to devise an ad-hoc algorithm for updating multipliers (very useful for UFLP).

Finding a feasible solution

Let us look at the Lagrangean Subproblem:

$$\min_{\underline{x},\underline{y}} \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} +$$

$$- \sum_{j \in J} \pi_j \left(\sum_{i \in I} x_{ij} - 1 \right) - \sum_{i \in I, j \in J} \eta_{ij} (y_i - x_{ij})$$

$$\text{s.t. } 0 \leq x_{ij} \leq 1 \qquad \forall i \in I, \forall j \in J$$

$$y_i \in \{0, 1\} \qquad \forall i \in I$$

$$u_{ij} \geq 0 \qquad \forall i \in I, \forall j \in J$$

Finding a feasible solution

The original problem is:

$$
\begin{aligned}
\min \quad & \sum_{i\in I} f_i y_i + \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} \\
\text{s.t.} \quad & \sum_{i\in I} x_{ij} = 1 && \forall j \in J \\
& x_{ij} \leq y_i && \forall i \in I, \forall j \in J \\
& x_{ij} \geq 0 && \forall i \in I, \forall j \in J \\
& y_i \in \{0,1\} && \forall i \in I
\end{aligned}
$$

and therefore, the Lagrangean Subproblem solution might violate

- assignment constraints
- (consistency) constraints linking x and y variables

how to build a feasible UFLP solution?

Finding a feasible solution

Some observations:

- once the y variables are fixed, optimizing over x is easy ...
- ... let's keep the values of y variables found in the Lagrangean Subproblem, and re-optimize over x.

(but we could do the other way round as well).

The idea of dual ascent

- If the Lagrangean subproblem has the integrality property, the corresponding lower bound is not tighter than that given by the continuous relaxation ...

The idea of dual ascent

- If the Lagrangean subproblem has the integrality property, the corresponding lower bound is not tighter than that given by the continuous relaxation ...
- ... but might be easier to compute!

The idea of dual ascent

- If the Lagrangean subproblem has the integrality property, the corresponding lower bound is not tighter than that given by the continuous relaxation ...
- ...but might be easier to compute!
- Subgradient algorithm is generic (not exploiting the structure of the problem).

The idea of dual ascent

- If the Lagrangean subproblem has the integrality property, the corresponding lower bound is not tighter than that given by the continuous relaxation ...
- ... but might be easier to compute!
- Subgradient algorithm is generic (not exploiting the structure of the problem).
- We can exploit LP theory and problem features to find a good set of multipliers.

Dual UFLP

- Let us consider the continuous relaxation of the inital UFLP formulation:

$$\begin{aligned}
\min \quad & \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} & \\
\text{s.t.} \quad & \sum_{i \in I} x_{ij} = 1 & \forall j \in J \\
& y_i - x_{ij} \geq 0 & \forall i \in I, \forall j \in J \\
& x_{ij} \geq 0 & \forall i \in I, \forall j \in J \\
& y_i \geq 0 & \forall i \in I
\end{aligned}$$

Università degli Studi di Milano

Dual UFLP

- Let us consider the continuous relaxation of the inital UFLP formulation:

$$\begin{aligned}
\min \; & \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} & \\
\text{s.t.} \; & \sum_{i \in I} x_{ij} = 1 & \forall j \in J \\
& y_i - x_{ij} \geq 0 & \forall i \in I, \forall j \in J \\
& x_{ij} \geq 0 & \forall i \in I, \forall j \in J \\
& y_i \geq 0 & \forall i \in I
\end{aligned}$$

- the corresponding dual problem is:

$$\begin{aligned}
\max \; & \sum_{j \in J} v_j & \\
\text{s.t.} \; & \sum_{j \in J} u_{ij} \leq f_i & \forall i \in I \\
& v_j - u_{ij} \leq c_{ij} & \forall i \in I, \forall j \in J \\
& u_{ij} \geq 0 & \forall i \in I, \forall j \in J
\end{aligned}$$

Dual UFLP

- Dual problem:

$$
\begin{aligned}
\max \ & \sum_{j \in J} v_j \\
\text{s.t. } & \sum_{j \in J} u_{ij} \leq f_i && \forall i \in I \\
& v_j - u_{ij} \leq c_{ij} && \forall i \in I, \forall j \in J \\
& u_{ij} \geq 0 && \forall i \in I, \forall j \in J
\end{aligned}
$$

UNIVERSITÀ DEGLI STUDI DI MILANO

Dual UFLP

- Dual problem:

$$\max \sum_{j \in J} v_j$$
$$\text{s.t.} \sum_{j \in J} u_{ij} \leq f_i \qquad \forall i \in I$$
$$v_j - u_{ij} \leq c_{ij} \qquad \forall i \in I, \forall j \in J$$
$$u_{ij} \geq 0 \qquad \forall i \in I, \forall j \in J$$

- For each choice of the dual variables v_j, fix u_{ij} to the maximum possible value (keep feasibility and leave objective function unchanged). Hence

$$u_{ij} = \max\{0, v_j - c_{ij}\}$$

Dual UFLP

- Dual problem:

$$\max \sum_{j \in J} v_j$$

$$\text{s.t.} \sum_{j \in J} u_{ij} \leq f_i \qquad \forall i \in I$$

$$v_j - u_{ij} \leq c_{ij} \qquad \forall i \in I, \forall j \in J$$

$$u_{ij} \geq 0 \qquad \forall i \in I, \forall j \in J$$

- For each choice of the dual variables v_j, fix u_{ij} to the maximum possible value (keep feasibility and leave objective function unchanged). Hence

$$u_{ij} = \max\{0, v_j - c_{ij}\}$$

- In this way we obtain a "condensed dual" problem:

$$\max \sum_{j \in J} v_j$$

$$\text{s.t.} \sum_{j \in J} \max\{0, v_j - c_{ij}\} \leq f_i \qquad \forall i \in I$$

DuaLoc

- Condensed dual problem:

$$\max \sum_{j \in J} v_j$$

$$\text{s.t.} \sum_{j \in J} \max\{0, v_j - c_{ij}\} \leq f_i \qquad \forall i \in I$$

DuaLoc

- Condensed dual problem:

$$\max \sum_{j \in J} v_j$$
$$\text{s.t.} \sum_{j \in J} \max\{0, v_j - c_{ij}\} \leq f_i \qquad \forall i \in I$$

- The DUALOC algorithm (Erlenkotter '78)
  - Inizialize $v_j = \min_{i \in I}\{c_{ij}\}$
  - Inizialize $s_i = f_i - \sum_{j \in J} \max\{0, v_j - c_{ij}\}$
  - Iteratively:
    - For each $j \in J$:
    - let $\Delta_j = \min_{i \in I}\{s_i | v_j - c_{ij} \geq 0\}$
    - decrease s_i of Δ_j for each i with $v_j - c_{ij} \geq 0$; increase v_j of Δ_j
  - Until there are no more changes in the dual solution.

Rebuild a primal solution

- Condensed dual problem:

$$\max \sum_{j \in J} v_j$$

$$\text{s.t.} \sum_{j \in J} \max\{0, v_j - c_{ij}\} \leq f_i \qquad \forall i \in I$$

Rebuild a primal solution

- Condensed dual problem:

$$\max \sum_{j \in J} v_j$$
$$\text{s.t.} \sum_{j \in J} \max\{0, v_j - c_{ij}\} \leq f_i \qquad \forall i \in I$$

- Complementary slackness conditions:

$$y_i(f_i - \sum_{j \in J} \max\{0, v_j - c_{ij}\}) = 0 \forall i \in I$$
$$(y_i - x_{ij}) \max\{0, v_j - c_{ij}\} = 0 \forall i \in I, \forall j \in J$$

Rebuild a primal solution

- Condensed dual problem:

$$\max \sum_{j \in J} v_j$$

$$\text{s.t.} \sum_{j \in J} \max\{0, v_j - c_{ij}\} \leq f_i \qquad \forall i \in I$$

- Complementary slackness conditions:

$$y_i(f_i - \sum_{j \in J} \max\{0, v_j - c_{ij}\}) = 0 \forall i \in I$$

$$(y_i - x_{ij}) \max\{0, v_j - c_{ij}\} = 0 \forall i \in I, \forall j \in J$$

- First, if $s_i > 0$ then let $y_i = 0$, else let $y_i = 1$

Rebuild a primal solution

- Condensed dual problem:

$$\max \sum_{j \in J} v_j$$
$$\text{s.t.} \sum_{j \in J} \max\{0, v_j - c_{ij}\} \leq f_i \qquad \forall i \in I$$

- Complementary slackness conditions:

$$y_i(f_i - \sum_{j \in J} \max\{0, v_j - c_{ij}\}) = 0 \forall i \in I$$
$$(y_i - x_{ij}) \max\{0, v_j - c_{ij}\} = 0 \forall i \in I, \forall j \in J$$

- First, if $s_i > 0$ then let $y_i = 0$, else let $y_i = 1$
- Second, assign each terminal j to a server in $\text{argmin}_{i \in I | y_i = 1} c_{ij}$

Rebuild a primal solution

- Condensed dual problem:

$$\max \sum_{j \in J} v_j$$
$$\text{s.t.} \sum_{j \in J} \max\{0, v_j - c_{ij}\} \le f_i \qquad \forall i \in I$$

- Complementary slackness conditions:

$$y_i(f_i - \sum_{j \in J} \max\{0, v_j - c_{ij}\}) = 0 \forall i \in I$$
$$(y_i - x_{ij}) \max\{0, v_j - c_{ij}\} = 0 \forall i \in I, \forall j \in J$$

- First, if $s_i > 0$ then let $y_i = 0$, else let $y_i = 1$
- Second, assign each terminal j to a server in $\text{argmin}_{i \in I | y_i = 1} c_{ij}$
- This condition might violate complementary slackness conditions, but is feasible, and therefore a valid *upper bound*.

FLP lab session

Implementing Lagrangean Relaxation and Column Generation FLP algorithms in AMPL.