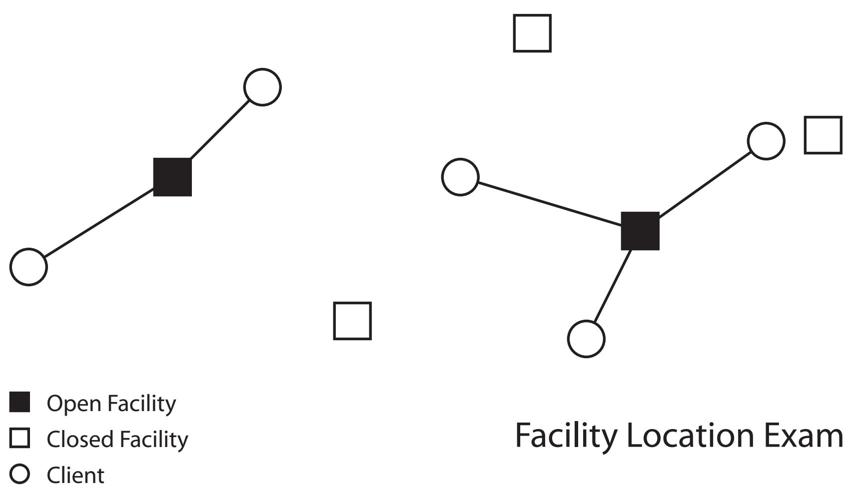
ORIE 6300 Mathematical Programming I	August 31, 2016
Recitation 2	
Lecturer: Calvin Wylie	*Topic: Facility Location*

Facility Location[1]

Problem Definition

In the facility location problem, we have a set of facilities $\mathcal{F}$ and a set of clients $\mathcal{C}$. The clients all have some demands that must be met. The facilities can service those demands, but only if they are open. Since the clients will always connect to the "closest" open facility, opening many facilities will allow clients to connect cheaply. However, there is also a cost associated with opening facilities, so opening fewer facilities at the cost of longer connections may be beneficial.



Facility Location Example

Formally, each facility $i \in \mathcal{F}$ has a cost f_i which is incurred if facility i is opened. For every facility i and client j, there is a connection cost $c_{ij} \geq 0$. Each client j must connect to a single facility i at cost c_{ij}. A solution is a subset S of the facilities $\mathcal{F}$ which will be opened, and the cost of this solution is

$$\sum_{i \in S} f_i + \sum_{j \in \mathcal{C}} \left(\min_{i \in S} c_{ij} \right).$$

We want to find a solution with minimal cost.

IP Formulation

Consider indicator ($\{0,1\}$) variables y_i, $i \in \mathcal{F}$ and x_{ij}, $(i,j) \in \mathcal{F} \times \mathcal{C}$ as representing whether we open facility i or whether we assign client j to facility i, respectively. This leads to a natural integer

[1]Based on previous notes of Chaoxu Tong

programming model given by:

$$\begin{array}{llrll}
\min & \sum_{i\in\mathcal{F}} f_i y_i & +\sum_{(i,j)\in\mathcal{F}\times\mathcal{C}} c_{i,j}x_{i,j} & & \\
\text{subject to:} & & \sum_{i\in\mathcal{F}} x_{i,j} & =1 & \forall j\in\mathcal{C} \\
& y_i & -x_{i,j} & \geq 0 & \forall (i,j)\in\mathcal{F}\times\mathcal{C} \\
& & y_i \in \{0,1\} & & \forall i\in\mathcal{F} \\
& & x_{i,j}\in\{0,1\} & & \forall (i,j)\in\mathcal{F}\times\mathcal{C}
\end{array}$$

The constraint $\sum_{i\in\mathcal{F}} x_{i,j} = 1$ represents that every client must be assigned to some facility while the constraint $y_i - x_{i,j} \geq 0$ represents that client j can only be assigned to facility i if we actually open facility i.

Calculating Lower Bounds

Just as with linear programs, we are often interested in calculating lower bounds on the objective value of integer programs. In this section, we will consider a scheme for deriving a family of lower bounds.

Initially, assume that all facility costs $f_i = 0$, i.e., there is no cost with opening a facility. Then, each client will choose to connect with the facility that has the lowest cost associated with it; hence, the effective cost for client j, denoted $v_j := \min_{i\in\mathcal{F}} c_{i,j}$, and the total objective cost is simply $\sum_{j\in\mathcal{C}} v_j$. Clearly, this is a lower bound on the optimal IP value.

Now, let's generalize this a little. Suppose that, instead of incurring the entire cost of opening a facility, each client has to pay specifically for the portion of a facility's resources it uses. More specifically, for each facility i, there are non-negative costs $w_{i,j}$ such that if client j is assigned to facility i, it must also pay $w_{i,j}$. Additionally, assume $\sum_{j\in\mathcal{C}} w_{i,j} \leq f_i$, i.e., the total of these individual costs is no more than the total cost of opening the facility.

To arrive at a lower bound, note that each client will want to minimize his total cost. Hence, each client's effective cost is now $v_j = \min_{i\in\mathcal{F}}\{c_{i,j} + w_{i,j}\}$. Once again, the corresponding lower bound on the objective cost is $\sum_{j\in\mathcal{C}} v_j$.

First, we can relax $v_j = \min_{i\in\mathcal{F}}\{c_{i,j} + w_{i,j}\}$ to $v_j \leq c_{i,j} + w_{i,j}\ \forall (i,j) \in \mathcal{F}\times\mathcal{C}$. Next, note that we never specified the actual costs, just simply that $\sum_{j\in\mathcal{C}} w_{i,j} \leq f_i$. So, maximize over the $w_{i,j}$ variables to get the best possible lower bound. This leads to an LP given by:

$$\begin{array}{llrll}
\max & \sum_{j\in\mathcal{C}} v_j & & & \\
\text{s.t.} & v_j & -w_{i,j} & \leq c_{i,j} & \forall (i,j)\in\mathcal{F}\times\mathcal{C} \\
& & \sum_{j\in\mathcal{C}} w_{i,j} & \leq f_i & \forall i\in\mathcal{F} \\
& & w_{i,j} & \geq 0 & \forall (i,j)\in\mathcal{F}\times\mathcal{C}
\end{array}$$

So, the problem of coming up with lower bounds for the facility location has been reduced to solving a linear program. By now, upon seeing a linear program, the first natural question to ask is what the dual represents.

Noting that there are $|\mathcal{F}\times\mathcal{C}|$ constraints in the first constraint set of the LP and $|\mathcal{F}|$ in the second, let $x_{i,j}$ and y_i be the corresponding dual variables. Then, the corresponding dual LP is :

$$\begin{array}{lllll}
\min & \sum_{(i,j)\in\mathcal{F}\times\mathcal{C}} c_{i,j}x_{i,j} & +\sum_{i\in\mathcal{F}} f_i y_i & & \\
\text{s.t.} & \sum_{i\in\mathcal{F}} x_{i,j} & & =1 & \forall j\in\mathcal{C} \\
 & -x_{i,j} & +y_i & \geq 0 & \forall (i,j)\in\mathcal{F}\times\mathcal{C} \\
 & x_{i,j}\geq 0 & y_i\geq 0 & & (i,j)\in\mathcal{F}\times\mathcal{C}
\end{array}$$

Note that this is exactly the LP relaxation of the original facility location problem. Hence, the method of combinatorially constructing lower bounds for our original problem corresponds directly with relaxing the integer program into a linear program and solving the dual.

Taking the Dual of an LP

Primal LP:
max $c^T x$
such that: $Ax = b$
$x \geq 0$

Dual LP:
min $b^T y$
such that: $A^T y \geq c$

Let's review the elements of dual construction. The dual is constructed to create a bound, so that the optimal value of the dual is greater than the optimal value of the primal. Proving this weak duality requires nothing more than matrix multiplication for some feasible x and y:

$$c^T x \leq y^T Ax = y^T b$$

Here, we have the first inequality because $x \geq 0$ and $A^T y \geq c$, and the second equality because $Ax = b$.

However, if for some row i of constraint A, we used the constraint $\sum_{j=1}^{n} a_{ij}x \leq b_i$, we would need an inequality for our variable y_i. Using $y_i \geq 0$ would guarantee that the direction of the inequality is maintained. On the other hand, if our primal used the constraint $\sum_{j=1}^{n} a_{ij}x \geq b_i$, we would need a different requirement for y_i. Specifically, $y_i \leq 0$ would ensure that the inequality flips so that the constraint $\sum_{j=1}^{n} a_{ij}x \geq b_i$ reverses direction, and can upper bound the primal LP value. Similarly, if we have some $x_j \leq 0$, then we require that for column j of A, $\sum_{i=1}^{m} a_{ij}y \leq c_j$ for the same reason. If x_j was unconstrained, we would instead require $\sum_{i=1}^{m} a_{ij}y = c_j$ because the sign of x_j is unknown.

This gives us the rules for nonstandard dual construction. There is a variable in the dual corresponding to each constraint in the primal, and there is a constraint in the dual corresponding to each variable in the primal. The relationships are as follows:

Primal (maximization)		Dual (minimization)
Constraint $\sum_{j=1}^{n} a_{ij}x \leq b_i$	$\Rightarrow$	Variable $y_i \geq 0$
Constraint $\sum_{j=1}^{n} a_{ij}x \geq b_i$	$\Rightarrow$	Variable $y_i \leq 0$
Constraint $\sum_{j=1}^{n} a_{ij}x = b_i$	$\Rightarrow$	Variable y_i unconstrained
Variable $x_j \geq 0$	$\Rightarrow$	Constraint $\sum_{i=1}^{m} a_{ij}y \geq c_j$
Variable $x_j \leq 0$	$\Rightarrow$	Constraint $\sum_{i=1}^{m} a_{ij}y \leq c_j$
Variable x_j unconstrained	$\Rightarrow$	Constraint $\sum_{i=1}^{m} a_{ij}y = c_j$