The Facility Location Problem

Zola Donovan

West Virginia University

April 21, 2014

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Metric uncapacitated facility location problem

Problem

1. Let G be a bipartite graph with bipartition (F, C), where F is the set of *facilities* and C is the set of *cities*.

Metric uncapacitated facility location problem

Problem

1. Let G be a bipartite graph with bipartition (F, C), where F is the set of *facilities* and C is the set of *cities*.
2. Let f_i be the cost of opening facility i, and c_{ij} be the cost of connecting city j to (opened) facility i.

Metric uncapacitated facility location problem

Problem

1. Let G be a bipartite graph with bipartition (F, C), where F is the set of *facilities* and C is the set of *cities*.
2. Let f_i be the cost of opening facility i, and c_{ij} be the cost of connecting city j to (opened) facility i.
3. The connection costs satisfy the triangle inequality.

Metric uncapacitated facility location problem

Problem

1. Let G be a bipartite graph with bipartition (F, C), where F is the set of *facilities* and C is the set of *cities*.
2. Let f_i be the cost of opening facility i, and c_{ij} be the cost of connecting city j to (opened) facility i.
3. The connection costs satisfy the triangle inequality.
4. The problem is to find a subset $I \subseteq F$ of facilities that should be opened,

Metric uncapacitated facility location problem

Problem

1. Let G be a bipartite graph with bipartition (F, C), where F is the set of *facilities* and C is the set of *cities*.
2. Let f_i be the cost of opening facility i, and c_{ij} be the cost of connecting city j to (opened) facility i.
3. The connection costs satisfy the triangle inequality.
4. The problem is to find a subset $I \subseteq F$ of facilities that should be opened, and a function $\phi : C \to I$ assigning cities to open facilities in such a way that the total cost of opening facilities and connecting cities to open facilities is minimized.

Integer program

Integer program

1. y_i is an indicator variable denoting whether facility i is open

Integer program

Integer program

1. y_i is an indicator variable denoting whether facility i is open
2. x_{ij} is an indicator variable denoting whether city j is connected to the facility i

Integer program

Integer program

1. y_i is an indicator variable denoting whether facility i is open
2. x_{ij} is an indicator variable denoting whether city j is connected to the facility i
3. The first set of constraints ensures that each city is connected to at least one facility

Integer program

Integer program

1. y_i is an indicator variable denoting whether facility i is open
2. x_{ij} is an indicator variable denoting whether city j is connected to the facility i
3. The first set of constraints ensures that each city is connected to at least one facility
4. The second set of constraints ensures that this facility must be open

Integer program

Integer program

1. y_i is an indicator variable denoting whether facility i is open
2. x_{ij} is an indicator variable denoting whether city j is connected to the facility i
3. The first set of constraints ensures that each city is connected to at least one facility
4. The second set of constraints ensures that this facility must be open

$$\text{minimize} \quad \sum_{i \in F, j \in C} c_{ij} \cdot x_{ij} + \sum_{i \in F} f_i \cdot y_i$$

Integer program

Integer program

1. y_i is an indicator variable denoting whether facility i is open
2. x_{ij} is an indicator variable denoting whether city j is connected to the facility i
3. The first set of constraints ensures that each city is connected to at least one facility
4. The second set of constraints ensures that this facility must be open

$$\begin{array}{rll} \text{minimize} & \sum_{i \in F, j \in C} c_{ij} \cdot x_{ij} + \sum_{i \in F} f_i \cdot y_i & \\ \text{subject to} & \sum_{i \in F} x_{ij} \geq 1, & j \in C \end{array}$$

Integer program

Integer program

1. y_i is an indicator variable denoting whether facility i is open
2. x_{ij} is an indicator variable denoting whether city j is connected to the facility i
3. The first set of constraints ensures that each city is connected to at least one facility
4. The second set of constraints ensures that this facility must be open

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i \in F, j \in C} c_{ij} \cdot x_{ij} + \sum_{i \in F} f_i \cdot y_i & \\
\text{subject to} \quad & \sum_{i \in F} x_{ij} \geq 1, & j \in C \\
& y_i - x_{ij} \geq 0, & i \in F, j \in C
\end{aligned}
$$

Integer program

Integer program

1. y_i is an indicator variable denoting whether facility i is open
2. x_{ij} is an indicator variable denoting whether city j is connected to the facility i
3. The first set of constraints ensures that each city is connected to at least one facility
4. The second set of constraints ensures that this facility must be open

$$
\begin{array}{lll}
\text{minimize} & \sum_{i \in F, j \in C} c_{ij} \cdot x_{ij} + \sum_{i \in F} f_i \cdot y_i & \\
\text{subject to} & \sum_{i \in F} x_{ij} \geq 1, & j \in C \\
& y_i - x_{ij} \geq 0, & i \in F, j \in C \\
& x_{ij} \in \{0, 1\}, & i \in F, j \in C
\end{array}
$$

Integer program

Integer program

1. y_i is an indicator variable denoting whether facility i is open
2. x_{ij} is an indicator variable denoting whether city j is connected to the facility i
3. The first set of constraints ensures that each city is connected to at least one facility
4. The second set of constraints ensures that this facility must be open

$$
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{i\in F, j\in C} c_{ij}\cdot x_{ij} + \sum_{i\in F} f_i\cdot y_i & \\
\text{subject to} & \displaystyle\sum_{i\in F} x_{ij} \geq 1, & j\in C \\
& y_i - x_{ij} \geq 0, & i\in F, j\in C \\
& x_{ij}\in\{0,1\}, & i\in F, j\in C \\
& y_i\in\{0,1\}, & i\in F
\end{array}
$$

LP-Relaxation of program

LP-Relaxation of program

$$\text{minimize} \quad \sum_{i \in F, j \in C} c_{ij} \cdot x_{ij} + \sum_{i \in F} f_i \cdot y_i$$

LP-Relaxation of program

LP-Relaxation of program

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in F, j \in C} c_{ij} \cdot x_{ij} + \sum_{i \in F} f_i \cdot y_i \\ \text{subject to} \quad & \sum_{i \in F} x_{ij} \geq 1, \qquad j \in C \end{aligned}$$

LP-Relaxation of program

LP-Relaxation of program

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{i\in F, j\in C} c_{ij}\cdot x_{ij} + \sum_{i\in F} f_i \cdot y_i & \\ \text{subject to} & \displaystyle\sum_{i\in F} x_{ij} \geq 1, & j \in C \\ & y_i - x_{ij} \geq 0, & i \in F, j \in C \end{array}$$

LP-Relaxation of program

LP-Relaxation of program

$$
\begin{array}{lrl}
\text{minimize} & \displaystyle\sum_{i \in F, j \in C} c_{ij} \cdot x_{ij} + \sum_{i \in F} f_i \cdot y_i & \\
\text{subject to} & \displaystyle\sum_{i \in F} x_{ij} \geq 1, & j \in C \\
 & y_i - x_{ij} \geq 0, & i \in F, j \in C \\
 & x_{ij} \geq 0, & i \in F, j \in C
\end{array}
$$

LP-Relaxation of program

LP-Relaxation of program

$$
\begin{array}{rll}
\text{minimize} & \sum_{i\in F, j\in C} c_{ij}\cdot x_{ij} + \sum_{i\in F} f_i \cdot y_i & \\
\text{subject to} & \sum_{i\in F} x_{ij} \geq 1, & j \in C \\
& y_i - x_{ij} \geq 0, & i\in F, j\in C \\
& x_{ij} \geq 0, & i\in F, j\in C \\
& y_i \geq 0, & i \in F
\end{array}
$$

Dual program

Dual program

Dual program

Dual program

$$\text{maximize} \qquad \sum_{j \in C} \alpha_j$$

Dual program

Dual program

$$
\begin{array}{lll}
\text{maximize} & \displaystyle\sum_{j \in C} \alpha_j & \\
\text{subject to} & \alpha_j - \beta_{ij} \leq c_{ij}, & i \in F, j \in C
\end{array}
$$

Dual program

Dual program

$$
\begin{array}{lll}
\text{maximize} & \sum_{j \in C} \alpha_j & \\
\text{subject to} & \alpha_j - \beta_{ij} \leq c_{ij}, & i \in F, j \in C \\
& \sum_{j \in C} \beta_{ij} \leq f_i, & i \in F
\end{array}
$$

Dual program

Dual program

$$
\begin{aligned}
\text{maximize} \quad & \sum_{j \in C} \alpha_j \\
\text{subject to} \quad & \alpha_j - \beta_{ij} \le c_{ij}, && i \in F, j \in C \\
& \sum_{j \in C} \beta_{ij} \le f_i, && i \in F \\
& \alpha_j \ge 0, && j \in C
\end{aligned}
$$

Dual program

Dual program

$$
\begin{array}{lll}
\text{maximize} & \displaystyle\sum_{j \in C} \alpha_j & \\
\text{subject to} & \alpha_j - \beta_{ij} \leq c_{ij}, & i \in F, j \in C \\
& \displaystyle\sum_{j \in C} \beta_{ij} \leq f_i, & i \in F \\
& \alpha_j \geq 0, & j \in C \\
& \beta_{ij} \geq 0, & i \in F, j \in C
\end{array}
$$

An intuitive understanding of the dual

Simple setting

1. Suppose LP has an optimal solution that is integral, say $I \subseteq F$ and $\phi : C \to I$.

An intuitive understanding of the dual

Simple setting

1. Suppose LP has an optimal solution that is integral, say $I \subseteq F$ and $\phi : C \to I$.
2. Then, under this solution, $y_i = 1$ iff $i \in I$, and $x_{ij} = 1$ iff $i = \phi(j)$.

An intuitive understanding of the dual

Simple setting

1. Suppose LP has an optimal solution that is integral, say $I \subseteq F$ and $\phi : C \to I$.
2. Then, under this solution, $y_i = 1$ iff $i \in I$, and $x_{ij} = 1$ iff $i = \phi(j)$.
3. Let (α, β) denote an optimal dual solution.

An intuitive understanding of the dual

The primal and dual complementary slackness conditions are:

(i) $\forall i \in F, j \in C : x_{ij} > 0 \Rightarrow \alpha_j - \beta_{ij} = c_{ij}$

An intuitive understanding of the dual

The primal and dual complementary slackness conditions are:

(i) $\forall i \in F, j \in C : x_{ij} > 0 \Rightarrow \alpha_j - \beta_{ij} = c_{ij}$

(ii) $\forall i \in F : y_i > 0 \Rightarrow \sum_{j \in C} \beta_{ij} = f_i$

An intuitive understanding of the dual

The primal and dual complementary slackness conditions are:

(i) $\forall i \in F, j \in C : x_{ij} > 0 \Rightarrow \alpha_j - \beta_{ij} = c_{ij}$

(ii) $\forall i \in F : y_i > 0 \Rightarrow \sum_{j \in C} \beta_{ij} = f_i$

(iii) $\forall j \in C : \alpha_j > 0 \Rightarrow \sum_{i \in F} x_{ij} = 1$

An intuitive understanding of the dual

The primal and dual complementary slackness conditions are:

(i) $\forall i \in F, j \in C : x_{ij} > 0 \Rightarrow \alpha_j - \beta_{ij} = c_{ij}$

(ii) $\forall i \in F : y_i > 0 \Rightarrow \sum_{j \in C} \beta_{ij} = f_i$

(iii) $\forall j \in C : \alpha_j > 0 \Rightarrow \sum_{i \in F} x_{ij} = 1$

(iv) $\forall i \in F, j \in C : \beta_{ij} > 0 \Rightarrow y_i = x_{ij}$

An intuitive understanding of the dual

The primal and dual complementary slackness conditions are:

(i) $\forall i \in F, j \in C : x_{ij} > 0 \Rightarrow \alpha_j - \beta_{ij} = c_{ij}$

(ii) $\forall i \in F : y_i > 0 \Rightarrow \sum_{j \in C} \beta_{ij} = f_i$

(iii) $\forall j \in C : \alpha_j > 0 \Rightarrow \sum_{i \in F} x_{ij} = 1$

(iv) $\forall i \in F, j \in C : \beta_{ij} > 0 \Rightarrow y_i = x_{ij}$

1. By condition (i), if $\phi(j) = i$, then $\alpha_j - \beta_{ij} = c_{ij}$.

An intuitive understanding of the dual

The primal and dual complementary slackness conditions are:

(i) $\forall i \in F, j \in C : x_{ij} > 0 \Rightarrow \alpha_j - \beta_{ij} = c_{ij}$

(ii) $\forall i \in F : y_i > 0 \Rightarrow \sum_{j \in C} \beta_{ij} = f_i$

(iii) $\forall j \in C : \alpha_j > 0 \Rightarrow \sum_{i \in F} x_{ij} = 1$

(iv) $\forall i \in F, j \in C : \beta_{ij} > 0 \Rightarrow y_i = x_{ij}$

1. By condition (i), if $\phi(j) = i$, then $\alpha_j - \beta_{ij} = c_{ij}$.
   So, α_j can be thought of as the total price paid by city j; where c_{ij} goes towards the use of edge (i,j), and β_{ij} is the contribution of j towards opening facility i.

An intuitive understanding of the dual

The primal and dual complementary slackness conditions are:

(i) $\forall i \in F, j \in C : x_{ij} > 0 \Rightarrow \alpha_j - \beta_{ij} = c_{ij}$

(ii) $\forall i \in F : y_i > 0 \Rightarrow \sum_{j \in C} \beta_{ij} = f_i$

(iii) $\forall j \in C : \alpha_j > 0 \Rightarrow \sum_{i \in F} x_{ij} = 1$

(iv) $\forall i \in F, j \in C : \beta_{ij} > 0 \Rightarrow y_i = x_{ij}$

1. By condition (i), if $\phi(j) = i$, then $\alpha_j - \beta_{ij} = c_{ij}$.
   So, α_j can be thought of as the total price paid by city j; where c_{ij} goes towards the use of edge (i,j), and β_{ij} is the contribution of j towards opening facility i.
2. By condition (ii), each open facility must be fully paid for, i.e., if $i \in I$, then $\sum_{j:\phi(j)=i} \beta_{ij} = f_i$.

An intuitive understanding of the dual

The primal and dual complementary slackness conditions are:

(i) $\forall i \in F, j \in C : x_{ij} > 0 \Rightarrow \alpha_j - \beta_{ij} = c_{ij}$

(ii) $\forall i \in F : y_i > 0 \Rightarrow \sum_{j \in C} \beta_{ij} = f_i$

(iii) $\forall j \in C : \alpha_j > 0 \Rightarrow \sum_{i \in F} x_{ij} = 1$

(iv) $\forall i \in F, j \in C : \beta_{ij} > 0 \Rightarrow y_i = x_{ij}$

1. By condition (i), if $\phi(j) = i$, then $\alpha_j - \beta_{ij} = c_{ij}$.
   So, α_j can be thought of as the total price paid by city j; where c_{ij} goes towards the use of edge (i, j), and β_{ij} is the contribution of j towards opening facility i.
2. By condition (ii), each open facility must be fully paid for, i.e., if $i \in I$, then $\sum_{j:\phi(j)=i} \beta_{ij} = f_i$.
3. Now consider condition (iv).

An intuitive understanding of the dual

The primal and dual complementary slackness conditions are:

(i) $\forall i \in F, j \in C : x_{ij} > 0 \Rightarrow \alpha_j - \beta_{ij} = c_{ij}$

(ii) $\forall i \in F : y_i > 0 \Rightarrow \sum_{j \in C} \beta_{ij} = f_i$

(iii) $\forall j \in C : \alpha_j > 0 \Rightarrow \sum_{i \in F} x_{ij} = 1$

(iv) $\forall i \in F, j \in C : \beta_{ij} > 0 \Rightarrow y_i = x_{ij}$

1. By condition (i), if $\phi(j) = i$, then $\alpha_j - \beta_{ij} = c_{ij}$.
   So, α_j can be thought of as the total price paid by city j; where c_{ij} goes towards the use of edge (i,j), and β_{ij} is the contribution of j towards opening facility i.
2. By condition (ii), each open facility must be fully paid for, i.e., if $i \in I$, then $\sum_{j:\phi(j)=i} \beta_{ij} = f_i$.
3. Now consider condition (iv). If facility i is open, but $\phi(j) \neq i$, then $y_i \neq x_{ij}$. This means $\beta_{ij} = 0$.

An intuitive understanding of the dual

The primal and dual complementary slackness conditions are:

(i) $\forall i \in F, j \in C : x_{ij} > 0 \Rightarrow \alpha_j - \beta_{ij} = c_{ij}$

(ii) $\forall i \in F : y_i > 0 \Rightarrow \sum_{j \in C} \beta_{ij} = f_i$

(iii) $\forall j \in C : \alpha_j > 0 \Rightarrow \sum_{i \in F} x_{ij} = 1$

(iv) $\forall i \in F, j \in C : \beta_{ij} > 0 \Rightarrow y_i = x_{ij}$

1. By condition (i), if $\phi(j) = i$, then $\alpha_j - \beta_{ij} = c_{ij}$.
   So, α_j can be thought of as the total price paid by city j; where c_{ij} goes towards the use of edge (i,j), and β_{ij} is the contribution of j towards opening facility i.
2. By condition (ii), each open facility must be fully paid for, i.e., if $i \in I$, then $\sum_{j:\phi(j)=i} \beta_{ij} = f_i$.
3. Now consider condition (iv). If facility i is open, but $\phi(j) \neq i$, then $y_i \neq x_{ij}$. This means $\beta_{ij} = 0$. In other words, city j does not contribute to opening any facility besides the one to which it is connected.

Relaxing primal complementary slackness conditions

Relaxing primal complementary slackness conditions

1. The cities are partitioned into two sets, *directly connected* and *indirectly connected*.

Relaxing primal complementary slackness conditions

Relaxing primal complementary slackness conditions

1. The cities are partitioned into two sets, *directly connected* and *indirectly connected*.
2. Only directly connected cities will pay for opening facilities, i.e., β_{ij} can be nonzero only if j is a directly connected city and $i = \phi(j)$.

Relaxing primal complementary slackness conditions

Relaxing primal complementary slackness conditions

1. The cities are partitioned into two sets, *directly connected* and *indirectly connected*.
2. Only directly connected cities will pay for opening facilities, i.e., β_{ij} can be nonzero only if j is a directly connected city and $i = \phi(j)$.
3. For an indirectly connected city j, the primal condition is relaxed as follows:

Relaxing primal complementary slackness conditions

Relaxing primal complementary slackness conditions

1. The cities are partitioned into two sets, *directly connected* and *indirectly connected*.
2. Only directly connected cities will pay for opening facilities, i.e., β_{ij} can be nonzero only if j is a directly connected city and $i = \phi(j)$.
3. For an indirectly connected city j, the primal condition is relaxed as follows:

$$(1/3)c_{\phi(j)j} \leq \alpha_j \leq c_{\phi(j)j}.$$

Relaxing primal complementary slackness conditions

Relaxing primal complementary slackness conditions

1. The cities are partitioned into two sets, *directly connected* and *indirectly connected*.
2. Only directly connected cities will pay for opening facilities, i.e., β_{ij} can be nonzero only if j is a directly connected city and $i = \phi(j)$.
3. For an indirectly connected city j, the primal condition is relaxed as follows:

$$(1/3)c_{\phi(j)j} \leq \alpha_j \leq c_{\phi(j)j}.$$

4. All other primal conditions are maintained:

Relaxing primal complementary slackness conditions

Relaxing primal complementary slackness conditions

1. The cities are partitioned into two sets, *directly connected* and *indirectly connected*.
2. Only directly connected cities will pay for opening facilities, i.e., β_{ij} can be nonzero only if j is a directly connected city and $i = \phi(j)$.
3. For an indirectly connected city j, the primal condition is relaxed as follows:

$$(1/3)c_{\phi(j)j} \leq \alpha_j \leq c_{\phi(j)j}.$$

4. All other primal conditions are maintained:

$\alpha_j - \beta_{\phi(j)j} = c_{\phi(j)j}$, for a directly connected city j,

Relaxing primal complementary slackness conditions

Relaxing primal complementary slackness conditions

1. The cities are partitioned into two sets, *directly connected* and *indirectly connected*.
2. Only directly connected cities will pay for opening facilities, i.e., β_{ij} can be nonzero only if j is a directly connected city and $i = \phi(j)$.
3. For an indirectly connected city j, the primal condition is relaxed as follows:
$$(1/3)c_{\phi(j)j} \leq \alpha_j \leq c_{\phi(j)j}.$$
4. All other primal conditions are maintained:
$\alpha_j - \beta_{\phi(j)j} = c_{\phi(j)j}$, for a directly connected city j,
and $\sum_{j:\phi(j)=i} \beta_{ij} = f_i$, for each open facility i.

Algorithm 1

Phase 1

In this phase, the algorithm:

Algorithm 1

Phase 1

In this phase, the algorithm:

1. operates in a primal-dual fashion

Algorithm 1

Phase 1

In this phase, the algorithm:

1. operates in a primal-dual fashion
2. finds a dual feasible solution

Algorithm 1

Phase 1

In this phase, the algorithm:

1. operates in a primal-dual fashion
2. finds a dual feasible solution
3. determines a set of tight edges and temporarily open facilities, F_t

Algorithm 1

Phase 1

In this phase, the algorithm:

1. operates in a primal-dual fashion
2. finds a dual feasible solution
3. determines a set of tight edges and temporarily open facilities, F_t

Phase 2

This phase consists of:

Algorithm 1

Phase 1

In this phase, the algorithm:

1. operates in a primal-dual fashion
2. finds a dual feasible solution
3. determines a set of tight edges and temporarily open facilities, F_t

Phase 2

This phase consists of:

1. choosing a subset I of F_t to open

Algorithm 1

Phase 1

In this phase, the algorithm:

1. operates in a primal-dual fashion
2. finds a dual feasible solution
3. determines a set of tight edges and temporarily open facilities, F_t

Phase 2

This phase consists of:

1. choosing a subset I of F_t to open
2. finding a mapping, ϕ, from cities to I

Algorithm 1

Phase 1

1. A notion of *time* is defined in this phase, so that each event can be associated with the time at which it happened; the phase starts at time 0.

Algorithm 1

Phase 1

1. A notion of *time* is defined in this phase, so that each event can be associated with the time at which it happened; the phase starts at time 0.
    1. Initially, each city is defined to be *unconnected*.

Algorithm 1

Phase 1

1. A notion of *time* is defined in this phase, so that each event can be associated with the time at which it happened; the phase starts at time 0.
   1. Initially, each city is defined to be *unconnected*.
2. Throughout this phase, the algorithm raises the dual variable α_j for each unconnected city j uniformly at unit rate, i.e., α_j will grow by 1 in unit time.

Algorithm 1

Phase 1

1. A notion of *time* is defined in this phase, so that each event can be associated with the time at which it happened; the phase starts at time 0.
    1. Initially, each city is defined to be *unconnected*.
2. Throughout this phase, the algorithm raises the dual variable α_j for each unconnected city j uniformly at unit rate, i.e., α_j will grow by 1 in unit time.
3. When $\alpha_j = c_{ij}$ for some edge (i, j), the algorithm will declare this edge to be *tight*.

Algorithm 1

Phase 1

1. A notion of *time* is defined in this phase, so that each event can be associated with the time at which it happened; the phase starts at time 0.
   1. Initially, each city is defined to be *unconnected*.
2. Throughout this phase, the algorithm raises the dual variable α_j for each unconnected city j uniformly at unit rate, i.e., α_j will grow by 1 in unit time.
3. When $\alpha_j = c_{ij}$ for some edge (i, j), the algorithm will declare this edge to be *tight*.
4. Henceforth, dual variable β_{ij} will be raised uniformly, thus ensuring that the first constraint in the LP is not violated.

Algorithm 1

Phase 1

1. A notion of *time* is defined in this phase, so that each event can be associated with the time at which it happened; the phase starts at time 0.
    1. Initially, each city is defined to be *unconnected*.
2. Throughout this phase, the algorithm raises the dual variable α_j for each unconnected city j uniformly at unit rate, i.e., α_j will grow by 1 in unit time.
3. When $\alpha_j = c_{ij}$ for some edge (i,j), the algorithm will declare this edge to be *tight*.
4. Henceforth, dual variable β_{ij} will be raised uniformly, thus ensuring that the first constraint in the LP is not violated. β_{ij} goes towards paying for facility i.

Algorithm 1

Phase 1

1. A notion of *time* is defined in this phase, so that each event can be associated with the time at which it happened; the phase starts at time 0.
    1. Initially, each city is defined to be *unconnected*.
2. Throughout this phase, the algorithm raises the dual variable α_j for each unconnected city j uniformly at unit rate, i.e., α_j will grow by 1 in unit time.
3. When $\alpha_j = c_{ij}$ for some edge (i,j), the algorithm will declare this edge to be *tight*.
4. Henceforth, dual variable β_{ij} will be raised uniformly, thus ensuring that the first constraint in the LP is not violated. β_{ij} goes towards paying for facility i. Each edge (i,j) such that $\beta_{ij} > 0$ is declared *special*.

Algorithm 1

Phase 1

1. A notion of *time* is defined in this phase, so that each event can be associated with the time at which it happened; the phase starts at time 0.
   1. Initially, each city is defined to be *unconnected*.
2. Throughout this phase, the algorithm raises the dual variable α_j for each unconnected city j uniformly at unit rate, i.e., α_j will grow by 1 in unit time.
3. When $\alpha_j = c_{ij}$ for some edge (i,j), the algorithm will declare this edge to be *tight*.
4. Henceforth, dual variable β_{ij} will be raised uniformly, thus ensuring that the first constraint in the LP is not violated. β_{ij} goes towards paying for facility i. Each edge (i,j) such that $\beta_{ij} > 0$ is declared *special*.
5. Facility i is said to be *paid for* if $\sum_j \beta_{ij} = f_i$. If so, the algorithm declares this facility *temporarily open*.

Algorithm 1

Phase 1

1. A notion of *time* is defined in this phase, so that each event can be associated with the time at which it happened; the phase starts at time 0.
   1. Initially, each city is defined to be *unconnected*.
2. Throughout this phase, the algorithm raises the dual variable α_j for each unconnected city j uniformly at unit rate, i.e., α_j will grow by 1 in unit time.
3. When $\alpha_j = c_{ij}$ for some edge (i,j), the algorithm will declare this edge to be *tight*.
4. Henceforth, dual variable β_{ij} will be raised uniformly, thus ensuring that the first constraint in the LP is not violated. β_{ij} goes towards paying for facility i. Each edge (i,j) such that $\beta_{ij} > 0$ is declared *special*.
5. Facility i is said to be *paid for* if $\sum_j \beta_{ij} = f_i$. If so, the algorithm declares this facility *temporarily open*.
   1. All unconnected cities having tight edges to this facility are declared *connected* and facility i is declared the *connecting witness* for each of these cities. (*Notice that dual variables α_j of these cities are no longer raised*.)

Algorithm 1

Phase 1

1. A notion of *time* is defined in this phase, so that each event can be associated with the time at which it happened; the phase starts at time 0.
    1. Initially, each city is defined to be *unconnected*.
2. Throughout this phase, the algorithm raises the dual variable α_j for each unconnected city j uniformly at unit rate, i.e., α_j will grow by 1 in unit time.
3. When $\alpha_j = c_{ij}$ for some edge (i, j), the algorithm will declare this edge to be *tight*.
4. Henceforth, dual variable β_{ij} will be raised uniformly, thus ensuring that the first constraint in the LP is not violated. β_{ij} goes towards paying for facility i. Each edge (i, j) such that $\beta_{ij} > 0$ is declared *special*.
5. Facility i is said to be *paid for* if $\sum_j \beta_{ij} = f_i$. If so, the algorithm declares this facility *temporarily open*.
    1. All unconnected cities having tight edges to this facility are declared *connected* and facility i is declared the *connecting witness* for each of these cities. (*Notice that dual variables α_j of these cities are no longer raised.*)
    2. In the future, as soon as an unconnected city j gets a tight edge to i, j will also be declared connected and i will be declared the connecting witness for j. (*Notice that $\beta_{ij} = 0$ and thus edge (i, j) is not special.*)

Algorithm 1

Phase 1

1. A notion of *time* is defined in this phase, so that each event can be associated with the time at which it happened; the phase starts at time 0.
    1. Initially, each city is defined to be *unconnected*.
2. Throughout this phase, the algorithm raises the dual variable α_j for each unconnected city j uniformly at unit rate, i.e., α_j will grow by 1 in unit time.
3. When $\alpha_j = c_{ij}$ for some edge (i, j), the algorithm will declare this edge to be *tight*.
4. Henceforth, dual variable β_{ij} will be raised uniformly, thus ensuring that the first constraint in the LP is not violated. β_{ij} goes towards paying for facility i. Each edge (i, j) such that $\beta_{ij} > 0$ is declared *special*.
5. Facility i is said to be *paid for* if $\sum_j \beta_{ij} = f_i$. If so, the algorithm declares this facility *temporarily open*.
    1. All unconnected cities having tight edges to this facility are declared *connected* and facility i is declared the *connecting witness* for each of these cities. (*Notice that dual variables α_j of these cities are no longer raised.*)
    2. In the future, as soon as an unconnected city j gets a tight edge to i, j will also be declared connected and i will be declared the connecting witness for j. (*Notice that $\beta_{ij} = 0$ and thus edge (i, j) is not special.*)
6. When all cities are connected, this phase of the algorithm terminates. If several events happen simultaneously, the algorithm executes them in arbitrary order.

Algorithm 1

Phase 2

1. Let F_t denote the set of temporarily open facilities and T denote the subgraph of G consisting of all special edges.

Algorithm 1

Phase 2

1. Let F_t denote the set of temporarily open facilities and T denote the subgraph of G consisting of all special edges.
    1. Let T^2 denote the graph that has edge (u, v) iff there is a path of length at most 2 between u and v in T, and let H be the subgraph of T^2 induced on F_t.

Algorithm 1

Phase 2

1. Let F_t denote the set of temporarily open facilities and T denote the subgraph of G consisting of all special edges.
   1. Let T^2 denote the graph that has edge (u, v) iff there is a path of length at most 2 between u and v in T, and let H be the subgraph of T^2 induced on F_t.
   2. Find any maximal independent set in H, say I.

Algorithm 1

Phase 2

1. Let F_t denote the set of temporarily open facilities and T denote the subgraph of G consisting of all special edges.
    1. Let T^2 denote the graph that has edge (u, v) iff there is a path of length at most 2 between u and v in T, and let H be the subgraph of T^2 induced on F_t.
    2. Find any maximal independent set in H, say I.
    3. All facilities in the set I are declared *open*.

Algorithm 1

Phase 2

1. Let F_t denote the set of temporarily open facilities and T denote the subgraph of G consisting of all special edges.
   1. Let T^2 denote the graph that has edge (u, v) iff there is a path of length at most 2 between u and v in T, and let H be the subgraph of T^2 induced on F_t.
   2. Find any maximal independent set in H, say I.
   3. All facilities in the set I are declared *open*.
2. For city j, define $\mathscr{F}_j = \{i \in F_t \mid (i, j) \text{ is special}\}$.

Algorithm 1

Phase 2

1. Let F_t denote the set of temporarily open facilities and T denote the subgraph of G consisting of all special edges.
   1. Let T^2 denote the graph that has edge (u, v) iff there is a path of length at most 2 between u and v in T, and let H be the subgraph of T^2 induced on F_t.
   2. Find any maximal independent set in H, say I.
   3. All facilities in the set I are declared *open*.
2. For city j, define $\mathscr{F}_j = \{i \in F_t \mid (i, j) \text{ is special}\}$.

Algorithm 1

Phase 2

1. Let F_t denote the set of temporarily open facilities and T denote the subgraph of G consisting of all special edges.
    1. Let T^2 denote the graph that has edge (u, v) iff there is a path of length at most 2 between u and v in T, and let H be the subgraph of T^2 induced on F_t.
    2. Find any maximal independent set in H, say I.
    3. All facilities in the set I are declared *open*.
2. For city j, define $\mathscr{F}_j = \{i \in F_t \mid (i, j) \text{ is special}\}$.
    1. Since I is an independent set, at most one of the facilities in $\mathscr{F}_j$ is opened.

Algorithm 1

Phase 2

1. Let F_t denote the set of temporarily open facilities and T denote the subgraph of G consisting of all special edges.
    1. Let T^2 denote the graph that has edge (u, v) iff there is a path of length at most 2 between u and v in T, and let H be the subgraph of T^2 induced on F_t.
    2. Find any maximal independent set in H, say I.
    3. All facilities in the set I are declared *open*.
2. For city j, define $\mathscr{F}_j = \{i \in F_t \mid (i, j) \text{ is special}\}$.
    1. Since I is an independent set, at most one of the facilities in $\mathscr{F}_j$ is opened.
    2. If there is a facility $i \in \mathscr{F}_j$ that is opened, then set $\phi(j) = i$ and declare city j *directly connected*.

Algorithm 1

Phase 2

1. Let F_t denote the set of temporarily open facilities and T denote the subgraph of G consisting of all special edges.
   1. Let T^2 denote the graph that has edge (u, v) iff there is a path of length at most 2 between u and v in T, and let H be the subgraph of T^2 induced on F_t.
   2. Find any maximal independent set in H, say I.
   3. All facilities in the set I are declared *open*.
2. For city j, define $\mathscr{F}_j = \{i \in F_t \mid (i, j) \text{ is special}\}$.
   1. Since I is an independent set, at most one of the facilities in $\mathscr{F}_j$ is opened.
   2. If there is a facility $i \in \mathscr{F}_j$ that is opened, then set $\phi(j) = i$ and declare city j *directly connected*.
   3. Otherwise, consider tight edge (i', j) such that i' was the connecting witness for j.

Algorithm 1

Phase 2

1. Let F_t denote the set of temporarily open facilities and T denote the subgraph of G consisting of all special edges.
    1. Let T^2 denote the graph that has edge (u, v) iff there is a path of length at most 2 between u and v in T, and let H be the subgraph of T^2 induced on F_t.
    2. Find any maximal independent set in H, say I.
    3. All facilities in the set I are declared *open*.
2. For city j, define $\mathscr{F}_j = \{i \in F_t \mid (i, j) \text{ is special}\}$.
    1. Since I is an independent set, at most one of the facilities in $\mathscr{F}_j$ is opened.
    2. If there is a facility $i \in \mathscr{F}_j$ that is opened, then set $\phi(j) = i$ and declare city j *directly connected*.
    3. Otherwise, consider tight edge (i', j) such that i' was the connecting witness for j.
    4. If $i' \in I$, again set $\phi(j) = i'$ and declare city j *directly connected*. (*Notice that in this case* $\beta_{i'j} = 0$).

Algorithm 1

Phase 2

1. Let F_t denote the set of temporarily open facilities and T denote the subgraph of G consisting of all special edges.
    1. Let T^2 denote the graph that has edge (u, v) iff there is a path of length at most 2 between u and v in T, and let H be the subgraph of T^2 induced on F_t.
    2. Find any maximal independent set in H, say I.
    3. All facilities in the set I are declared *open*.
2. For city j, define $\mathscr{F}_j = \{i \in F_t \mid (i, j) \text{ is special}\}$.
    1. Since I is an independent set, at most one of the facilities in $\mathscr{F}_j$ is opened.
    2. If there is a facility $i \in \mathscr{F}_j$ that is opened, then set $\phi(j) = i$ and declare city j *directly connected*.
    3. Otherwise, consider tight edge (i', j) such that i' was the connecting witness for j.
    4. If $i' \in I$, again set $\phi(j) = i'$ and declare city j *directly connected*. (*Notice that in this case* $\beta_{i'j} = 0$).
    5. In the remaining case that $i' \notin I$, let i be any neighbor of i' in graph H such that $i \in I$.

Algorithm 1

Phase 2

1. Let F_t denote the set of temporarily open facilities and T denote the subgraph of G consisting of all special edges.
   1. Let T^2 denote the graph that has edge (u, v) iff there is a path of length at most 2 between u and v in T, and let H be the subgraph of T^2 induced on F_t.
   2. Find any maximal independent set in H, say I.
   3. All facilities in the set I are declared *open*.
2. For city j, define $\mathscr{F}_j = \{i \in F_t \mid (i, j) \text{ is special}\}$.
   1. Since I is an independent set, at most one of the facilities in $\mathscr{F}_j$ is opened.
   2. If there is a facility $i \in \mathscr{F}_j$ that is opened, then set $\phi(j) = i$ and declare city j *directly connected*.
   3. Otherwise, consider tight edge (i', j) such that i' was the connecting witness for j.
   4. If $i' \in I$, again set $\phi(j) = i'$ and declare city j *directly connected*. (*Notice that in this case* $\beta_{i'j} = 0$).
   5. In the remaining case that $i' \notin I$, let i be any neighbor of i' in graph H such that $i \in I$.
   6. Set $\phi(j) = i$ and declare city j *indirectly connected*.

Algorithm 1

Phase 2

1. Let F_t denote the set of temporarily open facilities and T denote the subgraph of G consisting of all special edges.
   1. Let T^2 denote the graph that has edge (u, v) iff there is a path of length at most 2 between u and v in T, and let H be the subgraph of T^2 induced on F_t.
   2. Find any maximal independent set in H, say I.
   3. All facilities in the set I are declared *open*.
2. For city j, define $\mathscr{F}_j = \{i \in F_t \mid (i,j) \text{ is special}\}$.
   1. Since I is an independent set, at most one of the facilities in $\mathscr{F}_j$ is opened.
   2. If there is a facility $i \in \mathscr{F}_j$ that is opened, then set $\phi(j) = i$ and declare city j *directly connected*.
   3. Otherwise, consider tight edge (i', j) such that i' was the connecting witness for j.
   4. If $i' \in I$, again set $\phi(j) = i'$ and declare city j *directly connected*. (*Notice that in this case* $\beta_{i'j} = 0$).
   5. In the remaining case that $i' \notin I$, let i be any neighbor of i' in graph H such that $i \in I$.
   6. Set $\phi(j) = i$ and declare city j *indirectly connected*.
3. I and ϕ define primal integral solution: $x_{ij} = 1$ iff $\phi(j) = i$ and $y_i = 1$ iff $i \in I$.

Algorithm 1

Phase 2

1. Let F_t denote the set of temporarily open facilities and T denote the subgraph of G consisting of all special edges.
    1. Let T^2 denote the graph that has edge (u, v) iff there is a path of length at most 2 between u and v in T, and let H be the subgraph of T^2 induced on F_t.
    2. Find any maximal independent set in H, say I.
    3. All facilities in the set I are declared *open*.
2. For city j, define $\mathscr{F}_j = \{i \in F_t \mid (i, j) \text{ is special}\}$.
    1. Since I is an independent set, at most one of the facilities in $\mathscr{F}_j$ is opened.
    2. If there is a facility $i \in \mathscr{F}_j$ that is opened, then set $\phi(j) = i$ and declare city j *directly connected*.
    3. Otherwise, consider tight edge (i', j) such that i' was the connecting witness for j.
    4. If $i' \in I$, again set $\phi(j) = i'$ and declare city j *directly connected*. (*Notice that in this case* $\beta_{i'j} = 0$).
    5. In the remaining case that $i' \notin I$, let i be any neighbor of i' in graph H such that $i \in I$.
    6. Set $\phi(j) = i$ and declare city j *indirectly connected*.
3. I and ϕ define primal integral solution: $x_{ij} = 1$ iff $\phi(j) = i$ and $y_i = 1$ iff $i \in I$.
4. The values of α_j and β_{ij} obtained at the end of Phase 1 form a dual feasible solution.

Analysis

Analysis

1. We will show how the dual variables α_j's pay for the primal costs of opening facilities and connecting cities to facilities.

Analysis

Analysis

1. We will show how the dual variables α_j's pay for the primal costs of opening facilities and connecting cities to facilities.
2. Denote by α_j^f and α_j^e the contributions of city j to these two costs respectively; $\alpha_j = \alpha_j^f + \alpha_j^e$.

Analysis

Analysis

1. We will show how the dual variables α_j's pay for the primal costs of opening facilities and connecting cities to facilities.
2. Denote by α_j^f and α_j^e the contributions of city j to these two costs respectively; $\alpha_j = \alpha_j^f + \alpha_j^e$.
3. If j is indirectly connected, then $\alpha_j^f = 0$ and $\alpha_j^e = \alpha_j$

Analysis

Analysis

1. We will show how the dual variables α_j's pay for the primal costs of opening facilities and connecting cities to facilities.
2. Denote by α_j^f and α_j^e the contributions of city j to these two costs respectively; $\alpha_j = \alpha_j^f + \alpha_j^e$.
3. If j is indirectly connected, then $\alpha_j^f = 0$ and $\alpha_j^e = \alpha_j$
4. If j is directly connected, then the following must hold:

   $$\alpha_j = c_{ij} + \beta_{ij},$$

   where $i = \phi(j)$. Now, let $\alpha_j^f = \beta_{ij}$ and $\alpha_j^e = c_{ij}$.

Analysis

Lemma 1

Let $i \in I$. Then, $\sum_{j:\,\phi(j)=i} \alpha_j^f = f_i$.

Analysis

Lemma 1

Let $i \in I$. Then, $\sum_{j:\,\phi(j)=i} \alpha_j^f = f_i$.

Proof

1. Since i is temporarily open at the end of Phase 1, it is completely paid for, i.e.,

$$\sum_{j:\phi(j)=i} \alpha_j^f = f_i.$$

Analysis

Lemma 1

Let $i \in I$. Then, $\sum_{j:\phi(j)=i} \alpha_j^f = f_i$.

Proof

1. Since i is temporarily open at the end of Phase 1, it is completely paid for, i.e.,

$$\sum_{j:\phi(j)=i} \alpha_j^f = f_i.$$

2. Note that each city j that has contributed to f_i must be directly connected to i.

Analysis

Lemma 1

Let $i \in I$. Then, $\sum_{j:\phi(j)=i} \alpha_j^f = f_i$.

Proof

1. Since i is temporarily open at the end of Phase 1, it is completely paid for, i.e.,

$$\sum_{j:\phi(j)=i} \alpha_j^f = f_i.$$

2. Note that each city j that has contributed to f_i must be directly connected to i. For each such city, $\alpha_j^f = \beta_{ij}$.

Analysis

Lemma 1

Let $i \in I$. Then, $\sum_{j:\,\phi(j)=i} \alpha_j^f = f_i$.

Proof

1. Since i is temporarily open at the end of Phase 1, it is completely paid for, i.e.,

$$\sum_{j:\phi(j)=i} \alpha_j^f = f_i.$$

2. Note that each city j that has contributed to f_i must be directly connected to i. For each such city, $\alpha_j^f = \beta_{ij}$.
3. Any other city j' that is connected to facility i must satisfy $\alpha_{j'}^f = 0$.

Analysis

Lemma 1

Let $i \in I$. Then, $\sum_{j:\phi(j)=i} \alpha_j^f = f_i$.

Proof

1. Since i is temporarily open at the end of Phase 1, it is completely paid for, i.e.,
$$\sum_{j:\phi(j)=i} \alpha_j^f = f_i.$$
2. Note that each city j that has contributed to f_i must be directly connected to i. For each such city, $\alpha_j^f = \beta_{ij}$.
3. Any other city j' that is connected to facility i must satisfy $\alpha_{j'}^f = 0$.
4. The lemma follows. $\square$

Analysis

Corollary 2

$$\sum_{i \in I} f_i = \sum_{j \in C} \alpha_j^f.$$

Analysis

Corollary 2

$$\sum_{i \in I} f_i = \sum_{j \in C} \alpha_j^f.$$

Recall that α_j^f was defined to be 0 for indirectly connected cities. Thus, only the directly connected cities pay for the cost of opening facilities.

Analysis

Lemma 3

For an indirectly connected city j, $c_{ij} \leq 3\alpha_j^e$, where $i = \phi(j)$.

Analysis

Lemma 3

For an indirectly connected city $j, c_{ij} \leq 3\alpha_j^e$, where $i = \phi(j)$.

Proof

1. Let i' be the connecting witness for city j.

Analysis

Lemma 3

For an indirectly connected city $j, c_{ij} \leq 3\alpha_j^e$, where $i = \phi(j)$.

Proof

1. Let i' be the connecting witness for city j.
2. Since j is indirectly connected to i, (i, i') must be an edge in H.

Analysis

Lemma 3

For an indirectly connected city $j, c_{ij} \leq 3\alpha_j^e$, where $i = \phi(j)$.

Proof

1. Let i' be the connecting witness for city j.
2. Since j is indirectly connected to i, (i, i') must be an edge in H.
3. In turn, there must be a city, say j', such that (i, j') and (i', j') are both special edges.

Analysis

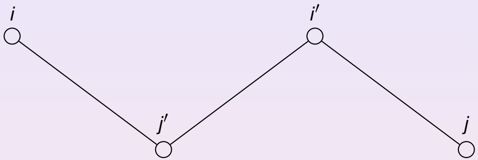
Lemma 3

For an indirectly connected city $j, c_{ij} \leq 3\alpha_j^e$, where $i = \phi(j)$.

Proof

1. Let i' be the connecting witness for city j.
2. Since j is indirectly connected to i, (i, i') must be an edge in H.
3. In turn, there must be a city, say j', such that (i, j') and (i', j') are both special edges.
4. Let t_1 and t_2 be the times at which i and i' were declared temporarily open during Phase 1.

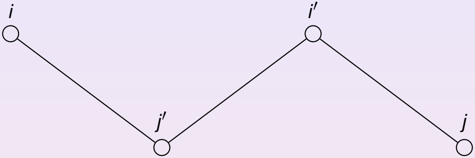
Analysis



Proof (*cont'd*)

1. Since edge (i', j) is tight, $\alpha_j \geq c_{i'j}$.

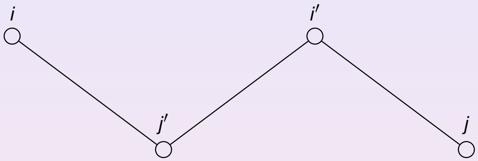
Analysis



Proof (*cont'd*)

1. Since edge (i', j) is tight, $\alpha_j \geq c_{i'j}$.
2. Since edges (i', j') and (i, j') are also tight, $\alpha_{j'} \geq c_{ij'}$ and $\alpha_{j'} \geq c_{i'j'}$.

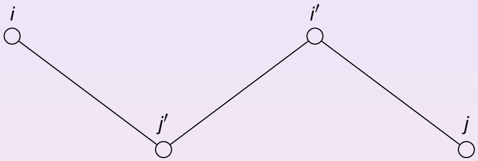
Analysis



Proof (*cont'd*)

1. Since edge (i', j) is tight, $\alpha_j \geq c_{i'j}$.
2. Since edges (i', j') and (i, j') are also tight, $\alpha_{j'} \geq c_{ij'}$ and $\alpha_{j'} \geq c_{i'j'}$.
3. Since both these edges are special, they must both have gone tight before either i or i' were declared temporarily open.

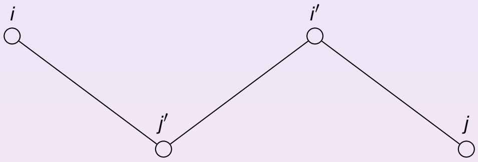
Analysis



Proof (*cont'd*)

1. Since edge (i', j) is tight, $\alpha_j \geq c_{i'j}$.
2. Since edges (i', j') and (i, j') are also tight, $\alpha_{j'} \geq c_{ij'}$ and $\alpha_{j'} \geq c_{i'j'}$.
3. Since both these edges are special, they must both have gone tight before either i or i' were declared temporarily open.
4. Consider the time $\min(t_1, t_2)$.

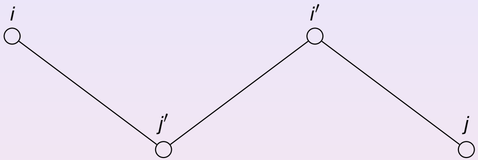
Analysis



Proof (*cont'd*)

1. Since edge (i', j) is tight, $\alpha_j \geq c_{i'j}$.
2. Since edges (i', j') and (i, j') are also tight, $\alpha_{j'} \geq c_{ij'}$ and $\alpha_{j'} \geq c_{i'j'}$.
3. Since both these edges are special, they must both have gone tight before either i or i' were declared temporarily open.
4. Consider the time $\min(t_1, t_2)$. $\alpha_{j'}$ cannot be growing beyond this time.

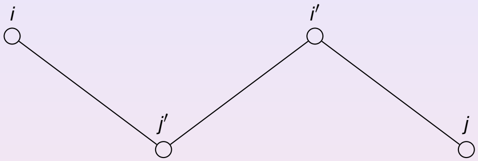
Analysis



Proof (*cont'd*)

1. Since edge (i', j) is tight, $\alpha_j \geq c_{i'j}$.
2. Since edges (i', j') and (i, j') are also tight, $\alpha_{j'} \geq c_{ij'}$ and $\alpha_{j'} \geq c_{i'j'}$.
3. Since both these edges are special, they must both have gone tight before either i or i' were declared temporarily open.
4. Consider the time $\min(t_1, t_2)$. $\alpha_{j'}$ cannot be growing beyond this time. Therefore, $\alpha_{j'} \leq \min(t_1, t_2)$.

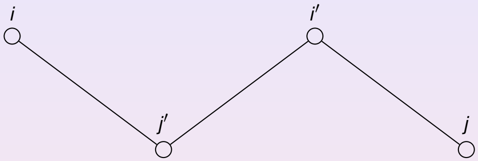
Analysis



Proof (*cont'd*)

1. Since edge (i', j) is tight, $\alpha_j \geq c_{i'j}$.
2. Since edges (i', j') and (i, j') are also tight, $\alpha_{j'} \geq c_{ij'}$ and $\alpha_{j'} \geq c_{i'j'}$.
3. Since both these edges are special, they must both have gone tight before either i or i' were declared temporarily open.
4. Consider the time $\min(t_1, t_2)$. $\alpha_{j'}$ cannot be growing beyond this time. Therefore, $\alpha_{j'} \leq \min(t_1, t_2)$.
5. Finally, since i' is the connecting witness for j, $\alpha_j \geq t_2$.

Analysis



Proof (*cont'd*)

1. Since edge (i', j) is tight, $\alpha_j \geq c_{i'j}$.
2. Since edges (i', j') and (i, j') are also tight, $\alpha_{j'} \geq c_{ij'}$ and $\alpha_{j'} \geq c_{i'j'}$.
3. Since both these edges are special, they must both have gone tight before either i or i' were declared temporarily open.
4. Consider the time $\min(t_1, t_2)$. $\alpha_{j'}$ cannot be growing beyond this time. Therefore, $\alpha_{j'} \leq \min(t_1, t_2)$.
5. Finally, since i' is the connecting witness for j, $\alpha_j \geq t_2$.
6. Therefore, $\alpha_j \geq \alpha_{j'}$, and the required inequalities follow. $\square$

Analysis

Theorem 4

The primal and dual solutions constructed by the algorithm satisfy:

$$\sum_{i\in F, j\in C} c_{ij}\cdot x_{ij} + 3\sum_{i\in F} f_i \cdot y_i \leq 3\sum_{j\in C}\alpha_j .$$

Analysis

Theorem 4

The primal and dual solutions constructed by the algorithm satisfy:

$$\sum_{i \in F, j \in C} c_{ij} \cdot x_{ij} + 3 \sum_{i \in F} f_i \cdot y_i \leq 3 \sum_{j \in C} \alpha_j.$$

Proof

For a directly connected city j, $c_{ij} = \alpha_j^e \leq 3\alpha_j^e, \textit{where } \phi(j) = i$.

Analysis

Theorem 4

The primal and dual solutions constructed by the algorithm satisfy:

$$\sum_{i\in F, j\in C} c_{ij}\cdot x_{ij} + 3\sum_{i\in F} f_i \cdot y_i \leq 3\sum_{j\in C}\alpha_j.$$

Proof

For a directly connected city j, $c_{ij} = \alpha_j^e \leq 3\alpha_j^e$, *where* $\phi(j) = i$.
Combined with Lemma 3, we get

$$\sum_{i\in F, j\in C} c_{ij}\cdot x_{ij} \leq 3\sum_{j\in C}\alpha_j^e.$$

Analysis

Theorem 4

The primal and dual solutions constructed by the algorithm satisfy:

$$\sum_{i \in F, j \in C} c_{ij} \cdot x_{ij} + 3 \sum_{i \in F} f_i \cdot y_i \leq 3 \sum_{j \in C} \alpha_j .$$

Proof

For a directly connected city j, $c_{ij} = \alpha_j^e \leq 3\alpha_j^e, where\ \phi(j) = i$.
Combined with Lemma 3, we get

$$\sum_{i \in F, j \in C} c_{ij} \cdot x_{ij} \leq 3 \sum_{j \in C} \alpha_j^e .$$

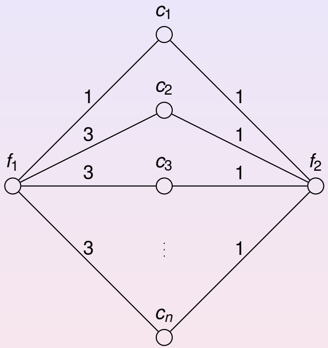
Adding this to the equality stated in Corollary 2, multiplied by 3, gives the theorem. $\square$

Analysis

Theorem 5

Algorithm 1 achieves an approximation factor of 3 for the facility location problem and has a running time of $O(m \log m)$.

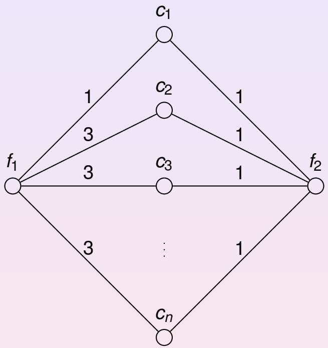
Tight example



Tight example

1. The graph has n cities, $c_1, c_2, \ldots, c_n$ and two facilities f_1 and f_2.

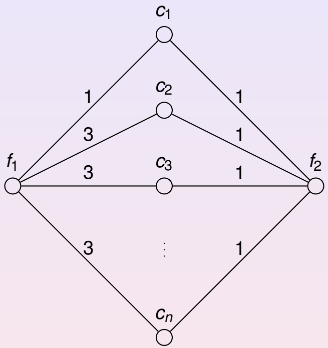
Tight example



Tight example

1. The graph has n cities, $c_1, c_2, \ldots, c_n$ and two facilities f_1 and f_2.
2. Each city is at a distance of 1 from f_2.

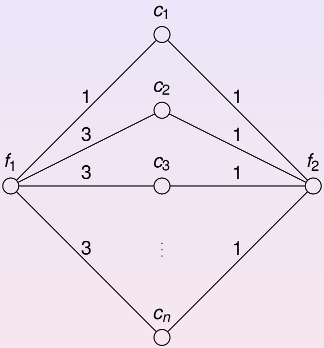
Tight example



Tight example

1. The graph has n cities, $c_1, c_2, \ldots, c_n$ and two facilities f_1 and f_2.
2. Each city is at a distance of 1 from f_2.
3. City c_1 is at a distance of 1 from f_1, and $c_2, \ldots, c_n$ are at a distance 3 from f_1.

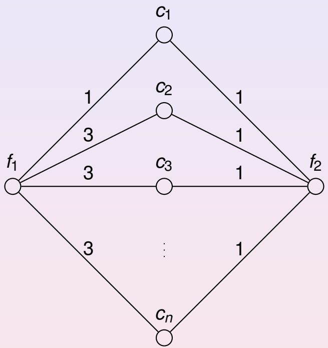
Tight example



Tight example

1. The graph has n cities, $c_1, c_2, \ldots, c_n$ and two facilities f_1 and f_2.
2. Each city is at a distance of 1 from f_2.
3. City c_1 is at a distance of 1 from f_1, and $c_2, \ldots, c_n$ are at a distance 3 from f_1.
4. The opening cost of f_1 and f_2 are ε and $(n+1)\varepsilon$, respectively, for a small number ε.

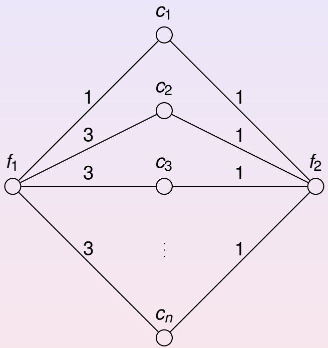
Tight example



Tight example (*cont'd*)

The optimal solution is to open f_2 and connect all cities to it, at a total cost of $(n+1)\varepsilon + n$.

Tight example



Tight example (*cont'd*)

The optimal solution is to open f_2 and connect all cities to it, at a total cost of $(n+1)\varepsilon + n$.

However, Algorithm 1 will open facility f_1 and connect all cities to it, at a total cost of $\varepsilon + 1 + 3(n-1)$.

References

Vazirani, V.V., 2003: *Approximation Algorithms*. Springer, 231 – 238.