Network Design

Part IV Tree problems

Minimum Spanning Tree

Given
A symmetric graph $G=(V,E)$ and a cost $c_e \geq 0$ for each edge in E
Find
A minimum cost spanning tree T

Notation
$E(S)$: set of edges induced by $S \subset V$
Cutset $\delta(S) = \{\{i,j\} \in E | i \in S, j \notin S\}$

Variables

$$x_e = \begin{cases} 1 \text{ if edge } e \text{ is selected,} \\ 0 \text{ otherwise.} \end{cases}$$

Constraints
T is spanning

$$\sum_{e \in E} x_e = |V| - 1$$

Minimum Spanning Tree

Connectivity
Can be imposed by one of the sets of constraints

Subtour inequalities	**Cutset inequalities**
$\sum_{e \in E(S)} x_e \leq \|S\| - 1$	$\sum_{e \in \delta(S)} x_e \geq 1$
$\forall S \subset V, 2 < \|S\| \leq \|V\| - 1$	$\forall S \subset V, S \neq \emptyset, V$

Observation
Both sets contains an exponential number (in $|V|$) of constraints

Minimum Spanning Tree

Formulation 1

$$
\begin{aligned}
\min \sum_{e \in E} c_e x_e & \\
\text{s.t.} \sum_{e \in E} x_e = |V| - 1, & \\
\sum_{e \in E(S)} x_e \leq |S| - 1 & \qquad \forall S \subset V, 2 < |S| \leq |V| - 1, \\
x \in \{0,1\}^{|E|} &
\end{aligned}
$$

Formulation 2

$$
\begin{aligned}
\min \sum_{e \in E} c_e x_e & \\
\text{s.t.} \sum_{e \in E} x_e = |V| - 1, & \\
\sum_{e \in \delta(S)} x_e \geq 1 & \qquad \forall S \subset V, S \neq \emptyset, V \\
x \in \{0,1\}^{|E|} &
\end{aligned}
$$

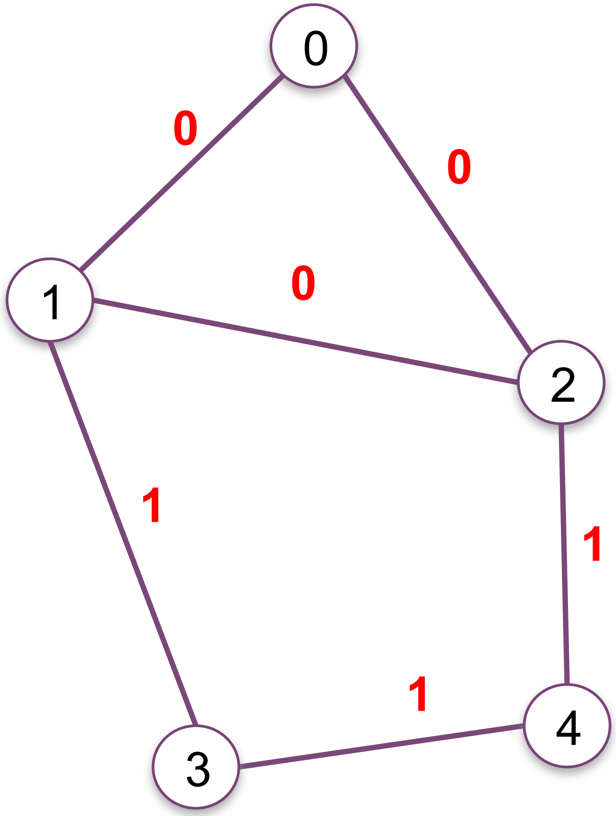
Results

Let P_{sub} be the polyhedron defined by the linear programming relaxation of Formulation 1 and P_{cut} be the polyhedron defined by the linear programming relaxation of Formulation 2. One can show that:

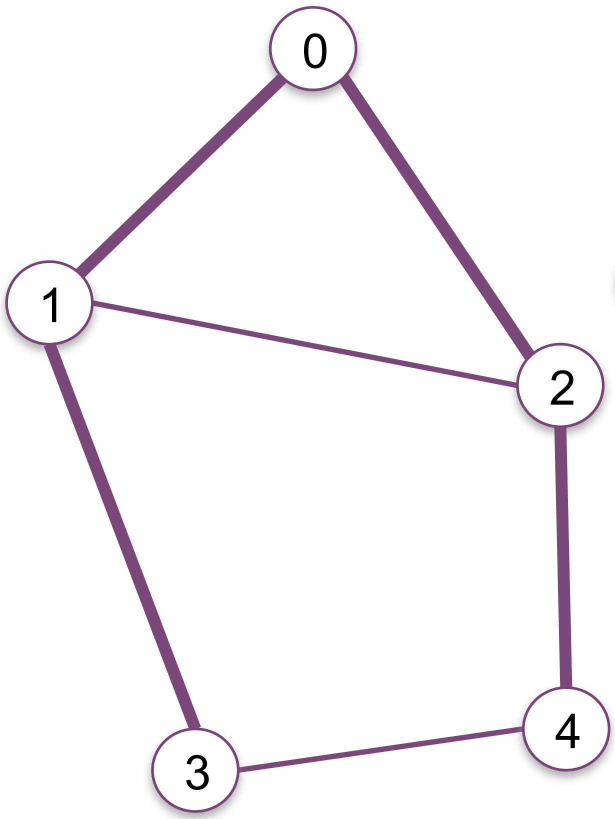
1. The extreme points of the polyhedron P_{sub} are the {0,1} incidence vectors of spanning trees
2. $P_{sub} \subseteq P_{cut}$
3. P_{cut} can have fractional extreme points

Example

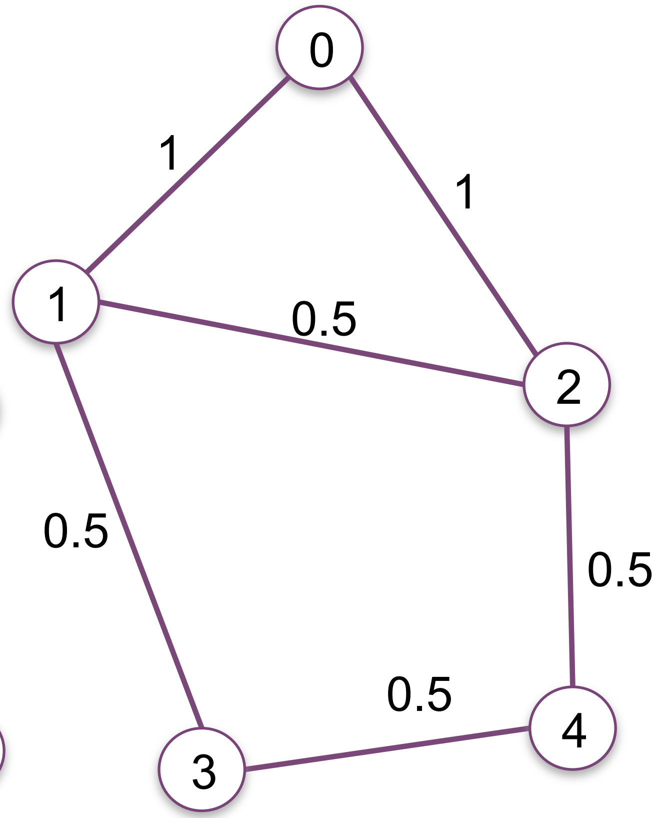
By solving P_{cut} on the graph with represented edge cost, one can get an optimal fractional solution



Edge costs



Optimal solution, z=2



Optimal **fractional** solution on P_{cut}
z=1.5

Code

The notebook **MST-all2017.ipynb** contains an implementation of Formulation 1.

The code **generator.py** invoked by the command

python generator.py 10 –d –n mygraph

returns a complete graph in graphML format embedded into a grid with edge costs equal to the euclidean distances between points in the grid

Constraint generation

Formulation 1 is impracticable even for small values of |V|
However one can resort to a “cutting plane” approach:

Algorithm

1. Initialize a formulation *P* with a subset (eventually empty) of Subtour Elimination Constraints

$$\sum_{e \in E(S)} x_e \leq |S| - 1 \text{ for some } S$$

2. Solve *P*
3. **If** the optimal solution x^*_{LP} of *P* satisfy all subtour inequalities then x^*_{LP} is also the optimal solution of P_{SUB}, **STOP**
   else find a set of nodes *S* such that

$$\sum_{e \in E(S)} x^*_e > |S| - 1$$

4. Add the constraint

$$\sum_{e \in E(S)} x_e \leq |S| - 1$$

to *P* and **GOTO 2**

Separation problem

The problem of finding a **violated Subtour Inequality** is called **separation problem** and can be formulated as an **optimization problem**

Decision variables

$$z_j \text{ for } j \in V, \text{ such that } \begin{cases} z_j = 1 \text{ if} j \in S \\ z_j = 0 \text{ otherwise.} \end{cases}$$

Separation problem

A subtour inequality is **violated** by x^*_{LP} if there exists a subset of nodes S such that

$$\sum_{e \in E(S)} x_e^* > |S| - 1$$

This is equivalent to:

$$\max_{S \subset V} \{ \sum_{e \in E(S)} x_e^* - |S| \} =$$

$$= \max_{S \subset V} \{ \sum_{e \in E(S)} x_e^* z_i z_j - \sum_{j \in V} z_j, e = \{i, j\} \} > -1$$

Separation problem

The optimal solution to

$$\max_{S \subset V} \{ \sum_{e \in E(S)} x_e^* z_i z_j - \sum_{j \in V} z_j , e = \{i, j\}\}$$

has value 0, with ***z***=0.
To avoid the trivial solution one has to fix $z_k = 1$ for k=1, …, $|V|$.

However, the problem needs to be linearized

Linearization

Consider the variable $w_{ij} = z_i \cdot z_j$

One has:

$$\max \sum_{e=\{i,j\}\in E} x_e^* w_{ij} - \sum_{j \in V} z_j$$

subject to

$$\forall e = \{i,j\} \in E : \begin{cases} w_{ij} - z_i \leq 0 \\ w_{ij} - z_j \leq 0 \\ w_{ij} - z_i - z_j \geq -1 \end{cases}$$

$$z_k = 1$$

$$z \in \{0,1\}^{|V|}, w \in \{0,1\}^{|E|}$$

Linearization

$$x_e^* \geq 0 \Rightarrow w_{ij} = \min\{z_i, z_j\}$$

$$\text{That is } \min\{z_i, z_j\} \geq z_i + z_j - 1$$

The separation problem reduces to

$$\max \sum_{e=\{i,j\}\in E} x_e^* w_{ij} - \sum_{j\in V} z_j$$

$$\text{subject to}$$

$$\forall e = \{i,j\} \in E : \begin{cases} w_{ij} - z_i \leq 0 \\ w_{ij} - z_j \leq 0 \end{cases}$$

$$z_k = 1$$

$$z \in [0,1]^{|V|}, w \in [0,1]^{|E|}$$

Note that integrality requirements can be dropped

Code

The notebook **MST-all2017.ipynb** contains an implementation of the whole procedure.

The code **generator.py** invoked by the command

python generator.py 10 –d –n mygraph

returns a complete graph in graphML format embedded into a grid with edge costs equal to the euclidean distances between points in the grid

Flow formulations

Single commodity flow formulation

Variables

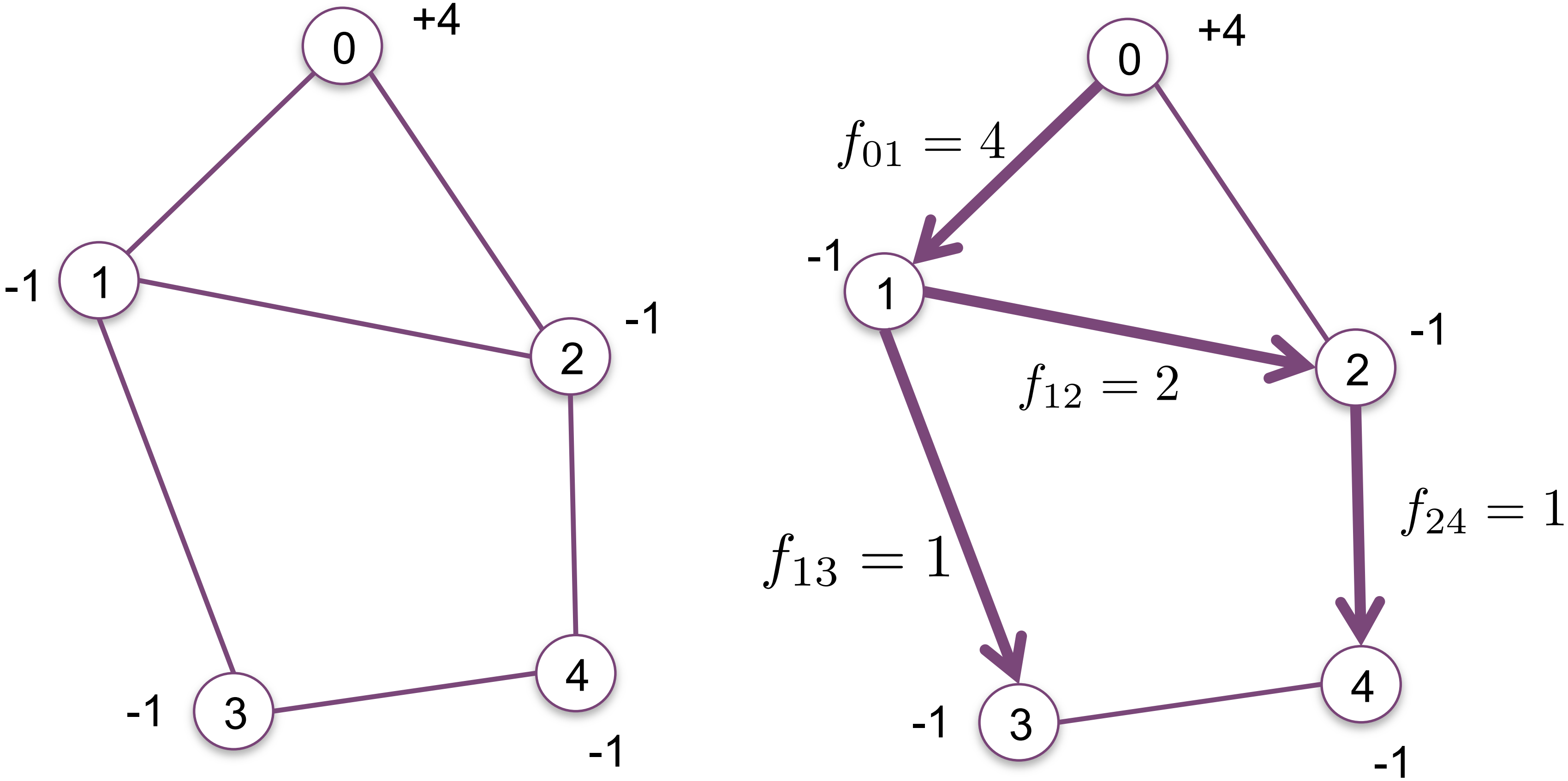
$$x_e = \begin{cases} 1 \text{ if edge } e \text{ is the tree} \\ 0 \text{ otherwise} \end{cases}$$

Associate to each edge *e* two directed arcs (*i*,*j*), (*j*,*i*)

$$f_{ij} = \{\text{Units of flow carried by arc } (i,j)\}$$

Separation

Single commodity flow formulation
The root node supplies *n*-1 units of flow to the other nodes that demand 1 unit of flow each



Minimum Spanning Tree

Single commodity flow formulation

$$\min cx$$

subject to:

$$\sum_{j \in \delta^+(0)} f_{0j} - \sum_{j \in \delta^-(0)} f_{j0} = n - 1$$

$$\sum_{j \in \delta^+(v)} f_{jv} - \sum_{j \in \delta^+(v)} f_{vj} = 1 \quad \forall v \in V, v \neq \{0\}$$

$$f_{ij} \leq (n-1)x_e \quad \forall e \in E, e = \{i, j\}$$

$$f_{ji} \leq (n-1)x_e \quad \forall e \in E, e = \{i, j\}$$

$$\sum_{e \in E} x_e = n - 1$$

$$f_{ij} \geq 0, x_e \in \{0, 1\}$$

Code

The notebook **MST-all2017.ipynb** contains an implementation of the single commodity flow formulation.

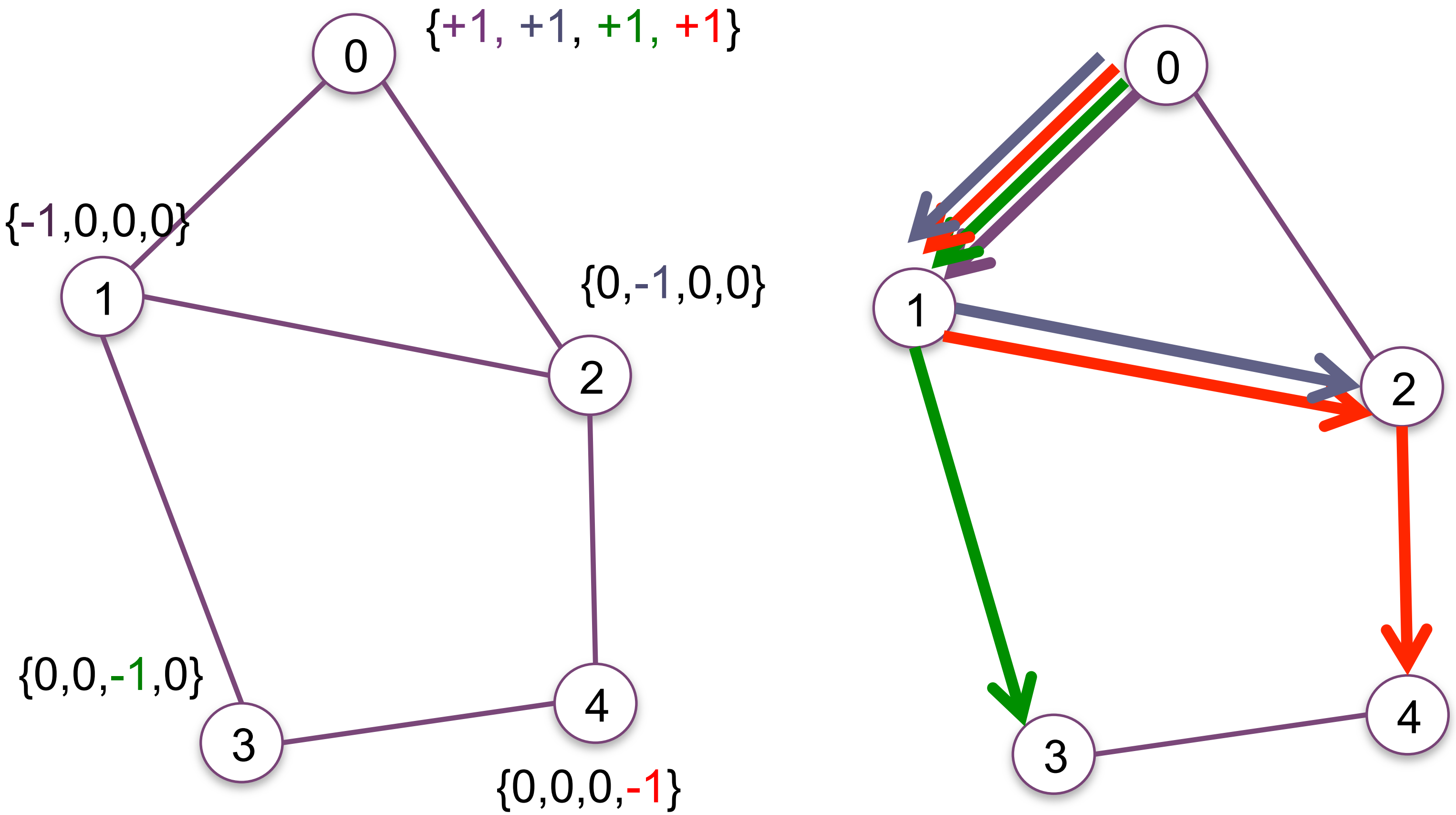
The code **generator.py** invoked by the command

python generator.py 10 –d –n mygraph

returns a complete graph in graphML format embedded into a grid with edge costs equal to the euclidean distances between points in the grid

Flow formulations

Directed Multicommodity flow formulation
The root node supplies n-1 commodities to the other nodes that demand 1 commodity each



Flow formulations

Multicommodity flow formulation

Variables

$$x_e = \begin{cases} 1 \text{ if edge } e \text{ is the tree} \\ 0 \text{ otherwise} \end{cases}$$

Associate to each edge *e* two directed arcs (*i*,*j*), (*j*,*i*)

$$y_{ij} = \text{ capacity for the flow of each commodity } k \text{ in arc } (i,j)$$

$$f_{ij}^k = \{\text{Flow of commodity } k \text{ carried by arc } (i,j)\}$$

Minimum Spanning Tree

Multicommodity flow formulation

$$\min cx$$

$$\text{subject to:}$$

$$\sum_{j \in \delta^+(0)} f_{0j}^k - \sum_{j \in \delta^-(0)} f_{j0}^k = 1 \quad \forall k \neq \{0\}$$

$$\sum_{j \in \delta^-(v)} f_{jv}^k - \sum_{j \in \delta^+(v)} f_{vj}^k = 0 \quad \forall k \neq \{0\}, \forall v \in V, v \neq \{0\}, v \neq k$$

$$\sum_{j \in \delta^-(k)} f_{jk}^k - \sum_{j \in \delta^+(k)} f_{kj}^k = 1 \quad \forall k \neq \{0\}$$

$$f_{ij}^k \leq y_{ij} \quad \forall (i,j) \text{ and } \forall k \neq \{0\}$$

$$\sum_{\{i,j\} \in E} (y_{ij} + y_{ji}) = n - 1$$

$$y_{ij} + y_{ji} = x_e \quad \forall (i,j) \in A, e = \{i,j\}$$

$$x_e \in \{0,1\} \quad \forall \{i,j\} \in E, y_{ij} \in \{0,1\} \quad \forall (i,j) \in A$$

$$f_{ij}^k \geq 0 \quad \forall (i,j) \in A, \forall k \neq \{0\}$$

Results

Let P_{FLOW} be the polyhedron defined by the linear programming relaxation of the Single commodity flow Formulation.

Let P_{DFLOW} be set of feasible solutions of the linear programming relaxation of the Multi commodity flow formulation in the *x*-space

1. P_{FLOW} contains fractional extreme points and the relaxation is generally weak

2. $P_{\text{DFLOW}} = P_{\text{SUB}}$

Both (single and multi) commodity flow formulations are compact

Code

The notebook **MST-multicommodityflow.ipynb** an implementation of of the multicommodity flow formulation.

The code **generator.py** invoked by the command

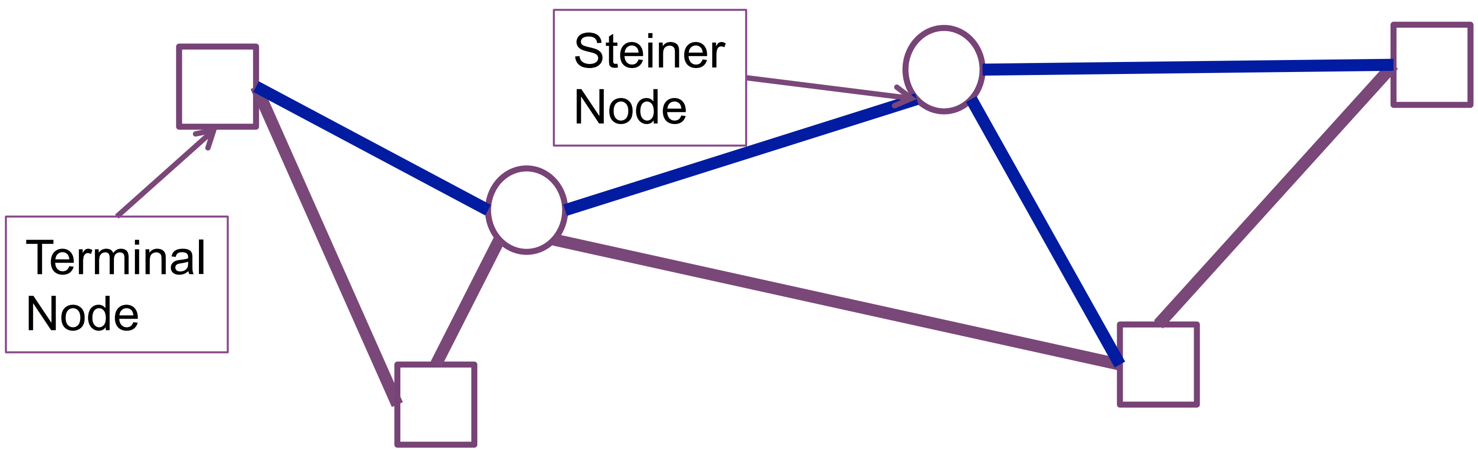
python generator.py 10 –d –n mygraph

returns a complete graph in graphML format embedded into a grid with edge costs equal to the euclidean distances between points in the grid

Steiner Tree

Given
A symmetric graph $G=(V,E)$ and a cost $c_e \geq 0$ for each edge in E
A set of terminal nodes $T \subset V$
Find
A minimum cost subtree spannig all terminal nodes $T \subset V$

The subtree might, or might not include some of the other optional "Steiner" nodes $S = V \setminus T$.



Prize Collecting Steiner Tree

Given

A symmetric graph $G=(V,E)$ and a cost $c_e \geq 0$ for each edge in E

A root node $\{0\}$

A profit $p_j > 0$ for each node j in $V \setminus \{0\}$

Find

A subtree T rooted in $\{0\}$ that maximizes the sum of the profits of the nodes in T minus the sum of the cost of the edges in T

Directed formulation

Consider the bidirected graph $B=(V,E)$ that is obtained from G by replacing each edge $e = \{i, j\}$ in E with two directed arcs (i, j) and
(j, i) (with corresponding weights $c_{ij} = c_{ji} = c_e$) and a cost $c_e \geq 0$ for each edge in E.

PCST is equivalent to find an optimal arborescence in B rooted in $\{0\}$

Variables

$$x_{ij} = \begin{cases} 1 \text{ if arc}(i,j) \text{ is in the arborescence} \\ 0 \text{ otherwise} \end{cases}$$

$$y_j = \begin{cases} 1 \text{ if node } j \text{ is in the arborescence} \\ 0 \text{ otherwise} \end{cases}$$

Directed Cut Formulation

$$
\begin{aligned}
\min \sum_{(i,j)\in A} c_{ij}x_{ij} - \sum_{j\in V\setminus\{0\}} y_j \\
\text{subject to} \\
y_0 &= 1 \\
\sum_{i\in\delta^-(j)} x_{ij} &= y_j \quad \forall j \in V \\
\sum_{(i,j)\in\delta^+(S)} x_{ij} &\geq y_k \quad \forall S \subset V, 0 \in S, k \in V \setminus S \\
x_{ij} &= \{0,1\} \;\; \forall (i,j) \in A \\
y_j &\in \{0,1\} \;\; \forall j \in V
\end{aligned}
$$

MTZ formulation

$u_j = \{$number of arcs in the dipath (if any) induced by x from $\{0\}$ to $j\}$

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij} - \sum_{j\in V\setminus\{0\}} p_j y_j$$

$$\text{subject to}$$

$$y_0 = 1$$

$$\sum_{i\in\delta^-(j)} x_{ij} = y_j \quad \forall j \in V \setminus \{0\}$$

$$(n+1)x_{ij} + u_i - u_j \leq n \quad \forall (i,j) \in A$$

$$x_{jk} \leq y_j \quad \forall j \in V \setminus \{0\}$$

$$0 \leq u_j \leq n \quad \forall j \in V$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in A$$

$$y_j \in \{0,1\} \quad \forall j \in V$$

MTZ formulation (2): lifting

$$\lambda_j = \{\text{value of the minimum possible number of arcs in a dipath in } B \text{ between}\{0\} \text{ and } j\}$$

The constraints MTZ_SEC

$$(n+1-\lambda_j)x_{ij} + (n-1-\lambda_j)x_{ji} + u_i - u_j \leq ny_i - \lambda_j y_j \ \ \forall (i,j) \in A$$

are valid subtour elimination constraints

MTZ formulation (3): lifting

The following inequalities are valid

$$\begin{cases} x_{ij} + x_{ji} \leq 1 \;\; \forall (i,j) \in A \\ \lambda_j \leq u_j \leq n, \;\; \forall j \in V \setminus \{0\} \end{cases}$$

Moreover:

$$u_j = 0 \text{ if } y_j = 0, \forall j \in V \setminus \{0\}$$

Thus, one has:

$$\lambda_j y_j \leq u_j \leq n y_j, \;\; \forall j \in V \setminus \{0\}$$

MTZ formulation (4): lifting

Case 1

If $x_{ij} = x_{ji} = 0$, since $u_i \leq n y_i$ and $u_j \geq \lambda_j y_j$

constraints MTZ_SEC are valid

Case 2

$x_{ij} = 1 \Rightarrow y_i = y_j = 1$ and $x_{ji} = 0$.

MTZ_SEC becomes $u_j \geq u_i + 1$ that is valid

Case 3

$x_{ji} = 1 \Rightarrow y_i = y_j = 1$ and $x_{ij} = 0$.

MTZ_SEC becomes $u_i \leq u_j + 1$.The symmetric inequality defined for the arc (i, j) is $u_i \geq u_j + 1$.

Thus, we get $u_i = u_j + 1$ that is valid.

Code

The code **pcst.py** contains an implementation of the MTZ formulation. The code **pcst_lifted.py** contains an implementation of the lifted MTZ formulation.

The code **generator.py** invoked by the command

python generator.py 10 –d –n mygraph –p 100-1000

returns a complete graph in graphML format embedded into a grid with edge costs equal to the euclidean distances between points in the grid and node profits in the range 100-1000

MTZ formulation (3)

$$\lambda_j = \{\text{value of the minimum possible number of arcs in a dipath in } B \text{ between} \{0\} \text{ and } j\}$$

$$\lambda_j y_j + \sum_{k \in \delta^-(j)} (\lambda_k - \lambda_j + 1) x_{kj} \leq u_j \leq n y_j - (n-1) x_{0j} \qquad \forall j \in V \setminus \{0\} : \{0\} \in \delta^-(j)$$

$$\lambda_j y_j + \sum_{k \in \delta^-(j)} (\lambda_k - \lambda_j + 1) x_{kj} \leq u_j \leq n y_j - \sum_{h \in \delta^+(j)} x_{jh} \qquad \forall j \in V \setminus \{0\}$$