A spanning tree heuristic for partitioning a graph into centered connected components

V. Gratta, I. Lari, J. Puerto, F. Ricca, A. Scozzari

Seville, December 15^{th} 2014

SAPIENZA
UNIVERSITÀ DI ROMA

p-Centered Partition Problem [Apollonio et al. 2008]

p-centered connected partition

Given a graph G=(V,E) and a subset S of vertices V called "centers", a p-centered partition is a partition into p connected components where each component contains exactly one center.

p-centered partition problem

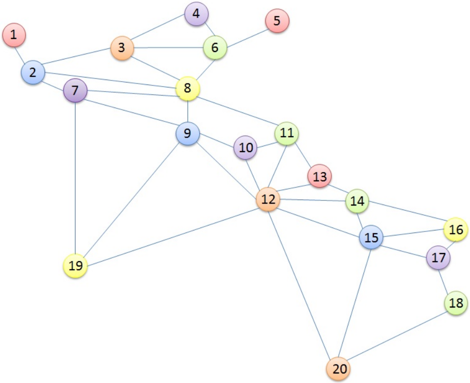
In the general p-centered partition problem we want to find a p-centered partition of the graph optimizing a cost-based objective function.

Application: clustering, image processing, territorial districting, etc. . .

Application: political districting [Ricca, Scozzari, Simeone, 2013]

Problem

Design a district map of the given territory, represented as a contiguity graph *([Simeone, 1978])*, and subdividing it into a fixed number of districts in which the election is performed.



1. p-Centered Partition Problem on graphs
   - Definition
   - Application: political districting
2. Problem on a tree T
   - Notation
   - Mathematical programming formulation
   - Solving
3. Future research
4. References

Notation

$T = (V, E)$	tree. $\|V\| = n$
$S \subseteq V$	centers. $\|S\| = p < n$
$U = V \setminus S$	units
$c : U \times S \to \mathbb{R}$	function that associates a cost $c_{is} \geq 0$ to each pair $(i, s), i \in U, s \in S$

Problem

Find a p-centered partition of T that minimizes the maximum assignment cost of a unit $i \in U$ to a center $s \in S$.

Mathematical programming formulation on T

Variables

$$y_{is} = \begin{cases} 1 & \text{if unit } i \text{ is assigned to center } s \\ 0 & \text{otherwise} \end{cases} \qquad i \in U, s \in S$$

Mathematical programming formulation on T

Constraints

$j(i,s)$: vertex j that is adjacent to i in the unique path from i to s in T

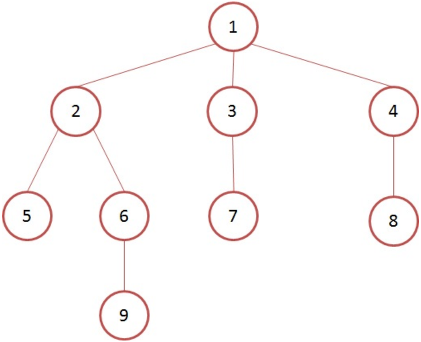
$$y_{is} \leq y_{j(i,s)s} \qquad \forall i \in U, s \in S, (i,s) \notin E$$

Mathematical programming formulation on T

Constraints

$j(i,s)$: vertex j that is adjacent to i in the unique path from i to s in T

$$y_{is} \leq y_{j(i,s)s} \qquad \forall i \in U, s \in S, (i,s) \notin E$$

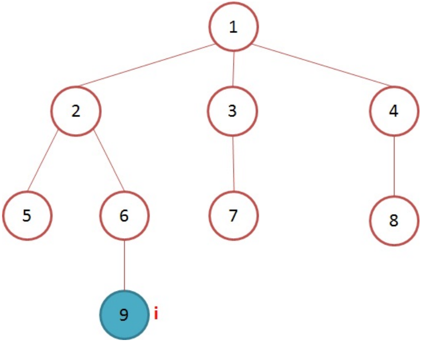


Mathematical programming formulation on T

Constraints

$j(i,s)$: vertex j that is adjacent to i in the unique path from i to s in T

$$y_{is} \leq y_{j(i,s)s} \qquad \forall i \in U, s \in S, (i,s) \notin E$$

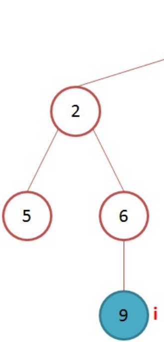


Mathematical programming formulation on T

Constraints

$j(i,s)$: vertex j that is adjacent to i in the unique path from i to s in T

$$y_{is} \leq y_{j(i,s)s} \qquad \forall i \in U, s \in S, (i,s) \notin E$$

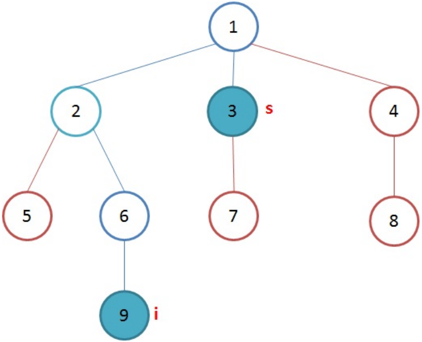


Mathematical programming formulation on T

Constraints

$j(i,s)$: vertex j that is adjacent to i in the unique path from i to s in T

$$y_{is} \leq y_{j(i,s)s} \qquad \forall i \in U, s \in S, (i,s) \notin E$$

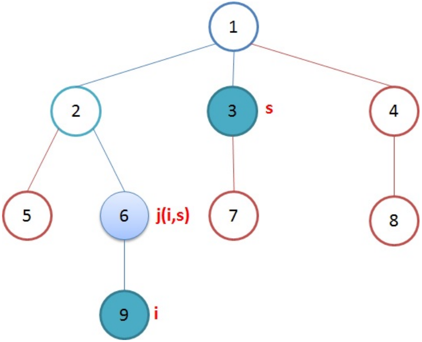


Mathematical programming formulation on T

Constraints

$j(i,s)$: vertex j that is adjacent to i in the unique path from i to s in T

$$y_{is} \leq y_{j(i,s)s} \qquad \forall i \in U, s \in S, (i,s) \notin E$$



Mathematical programming formulation on T

Constraints

$$\sum_{s \in S} y_{is} = 1 \qquad \forall i \in U$$

Each unit i must be assigned to exactly one center s.

Mathematical programming formulation on T

Constraints

$$\sum_{s \in S} y_{is} = 1 \qquad \forall i \in U$$

Each unit i must be assigned to exactly one center s.

Objective function

$$\min \max_{s \in S} \max_{i \in U} c_{is} y_{is}$$

Minimize of the worst-case assigning cost.

Mathematical programming formulation on T

$$
\begin{array}{llcll}
\min & \max\limits_{s\in S} \max\limits_{i\in U} & & c_{is}y_{is} & \\
s.t. & y_{is} & \leq & y_{j(i,s)s} & \forall i\in U, s\in S, (i,s)\notin E \\
 & \sum\limits_{s\in S} y_{is} & = & 1 & \forall i \in U \\
 & y_{is} & \in & \{0,1\} & \forall i\in U, s\in S
\end{array} \tag{1}
$$

Mathematical programming formulation (Feasibility Problem)

Given a fixed value α, find, if exists, a p-centered partition of T such that $\max_{s \in S} \max_{i \in U} c_{is} y_{is} \leq \alpha$

$$\begin{array}{rcll} y_{is} & \leq & y_{j(i,s)s} & \forall i \in U, s \in S, (i,s) \notin E \\ \sum_{s \in S} y_{is} & = & 1 & \forall i \in U \\ y_{is} & \in & \{0,1\} & \forall i \in U, s \in S \\ y_{is} & = & 0 & if \ \ c_{is} > \alpha, i \in U, s \in S \end{array} \quad (2)$$

Mathematical programming formulation (Relaxed Feasibility Problem)

Given a fixed value α, find, if exists, a p-centered partition of T such that $\max_{s \in S} \max_{i \in U} c_{is} y_{is} \leq \alpha$

$$
\begin{array}{rcll}
y_{is} & \leq & y_{j(i,s)s} & \forall i \in U, s \in S, (i,s) \notin E \\
\sum_{s \in S} y_{is} & = & 1 & \forall i \in U \\
y_{is} & \geq & 0 & \forall i \in U, s \in S \\
y_{is} & = & 0 & if \;\; c_{is} > \alpha, i \in U, s \in S
\end{array}
\tag{3}
$$

This is a Linear Programming problem and his feasible polytope has integer vertices ([Lari, Puerto, Ricca, Scozzari, 2014]).

Algorithm [Lari, Puerto, Ricca, Scozzari, 2014]

1. Sort the c_{is} values, $i \in U, s \in S$, in non-decreasing order
2. Apply a binary search to generate all the possible different values $\alpha = \min \max_{s \in S} \max_{i \in U} c_{is} y_{is}$ of problem (1).
3. For each α solve the feasibility problem (3).

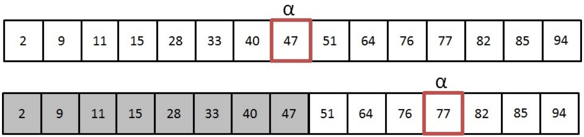
Algorithm [Lari, Puerto, Ricca, Scozzari, 2014]

1. Sort the c_{is} values, $i \in U, s \in S$, in non-decreasing order
2. Apply a binary search to generate all the possible different values $\alpha = \min \max_{s \in S} \max_{i \in U} c_{is} y_{is}$ of problem (1).
3. For each α solve the feasibility problem (3).

							α							
2	9	11	15	28	33	40	47	51	64	76	77	82	85	94

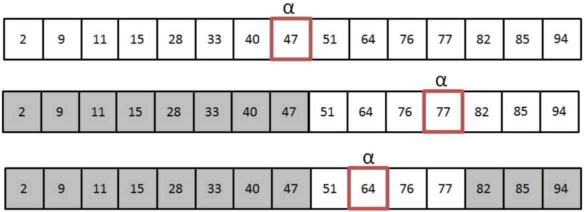
Algorithm

1. Sort the c_{is} values, $i \in U, s \in S$, in non-decreasing order
2. Apply a binary search to generate all the possible different values $\alpha = \min_{s \in S} \max_{i \in U} \max c_{is} y_{is}$ of the problem (1).
3. For each α solve the feasibility problem (3).



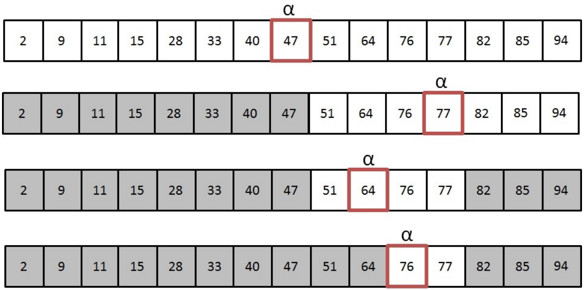
Algorithm

1. Sort the c_{is} values, $i \in U, s \in S$, in non-decreasing order
2. Apply a binary search to generate all the possible different values $\alpha = \min_{s \in S} \max_{i \in U} c_{is} y_{is}$ of the problem (1).
3. For each α solve the feasibility problem (3).



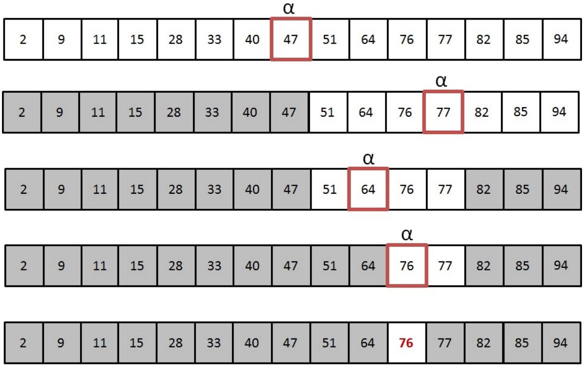
Algorithm

1. Sort the c_{is} values, $i \in U, s \in S$, in non-decreasing order
2. Apply a binary search to generate all the possible different values $\alpha = \min_{s \in S} \max_{i \in U} \max c_{is} y_{is}$ of the problem (1).
3. For each α solve the feasibility problem (3).



Algorithm

1. Sort the c_{is} values, $i \in U, s \in S$, in non-decreasing order
2. Apply a binary search to generate all the possible different values $\alpha = \min_{s \in S} \max_{i \in U} \max c_{is} y_{is}$ of the problem (1).
3. For each α solve the feasibility problem (3).



Problem on a graph

We know that the problem is NP-hard on general graphs, but we have a polynomial time algorithms for trees ([Lari, Puerto, Ricca, Scozzari, 2014]).

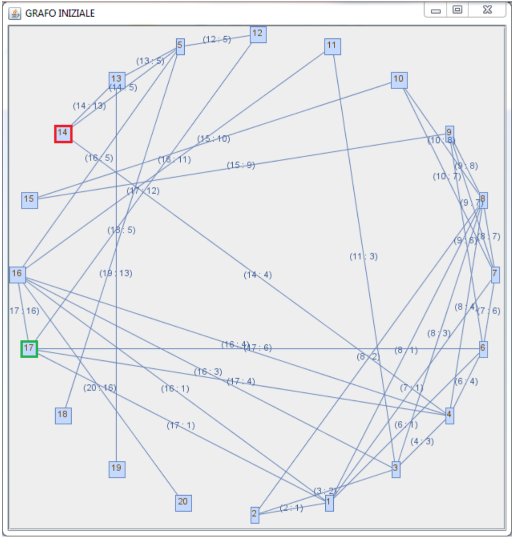
Idea

Exploit the exact algorithm on trees to solve heuristically the problem on general graphs, basing on the correspondence that exists between the optimal partition of a graph and that of one of its spanning trees. ([Maravalle e Simeone, 1995]).

Basic idea of the heuristic algorithm

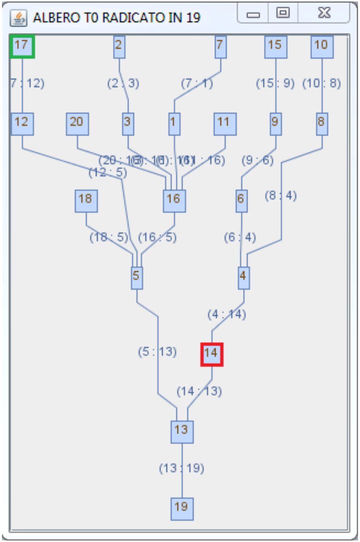
1. Generate a spanning tree of G, $T = (V, E_T)$.
2. Apply to T the polynomial time algorithm for trees.
3. Modify locally T to obtain a new spanning tree of G, $T' \neq T$
4. Update T with T' and go to 2

Example

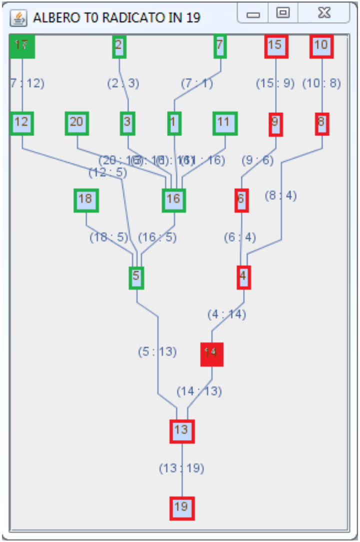


$$C = \begin{pmatrix} 6 & 13 \\ 6 & 6 \\ 9 & 10 \\ 11 & 12 \\ 5 & 4 \\ 1 & 4 \\ 14 & 8 \\ 10 & 1 \\ 5 & 4 \\ 6 & 8 \\ 4 & 7 \\ 9 & 14 \\ 8 & 8 \\ 6 & 6 \\ 2 & 1 \\ 13 & 7 \\ 7 & 6 \\ 11 & 13 \end{pmatrix}$$

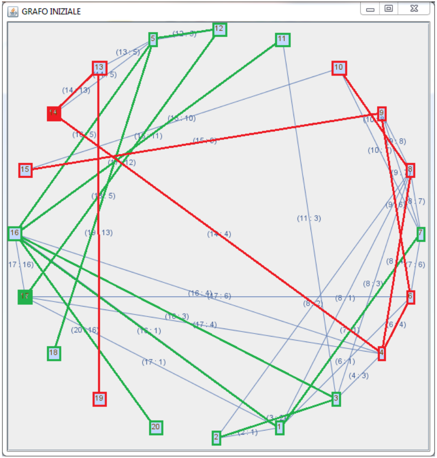
Example


ALBERO T0 RADICATO IN 19
17
2
7
15
10
12
20
3
1
11
9
8
18
16
6
5
4
14
13
19
(13 : 19)
(14 : 13)
(5 : 13)
(4 : 14)

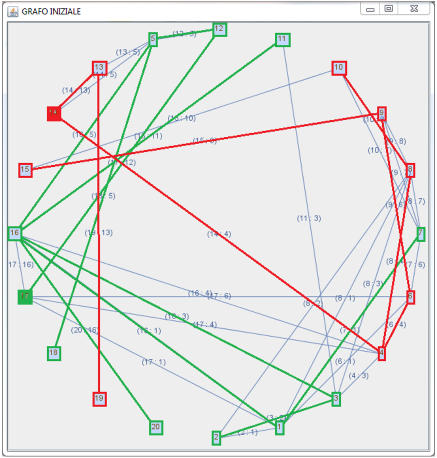
Example

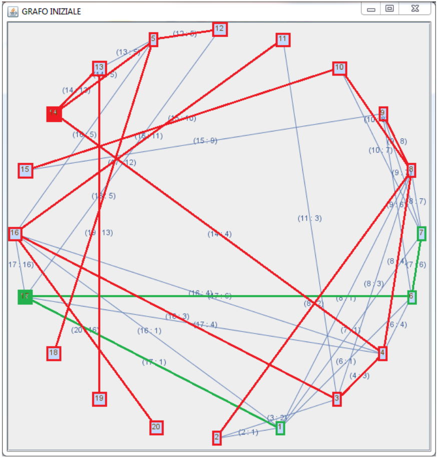

ALBERO T0 RADICATO IN 19
17
2
7
15
10
(2 : 3)
(7 : 1)
(15 : 9)
(10 : 8)
12
20
3
1
11
9
8
(9 : 6)
(12 : 5)
18
16
6
(8 : 4)
(18 : 5)
(16 : 5)
(6 : 4)
5
4
(4 : 14)
(5 : 13)
14
(14 : 13)
13
(13 : 19)
19

Example



Example





References

I. Lari, J. Puerto, F. Ricca, A. Scozzari, 2014
Partitioning a graph into connected components with fixed centers and optimizing different criteria.
submitted to the scientific journal *Networks.*

F. Ricca, A. Scozzari, B. Simeone, 2011
Political districting: from classical models to recent approaches.
Annals of Operations Research, *vol 204 pp 271–299 2013.*

N. Apollonio, I. Lari, J. Puerto, F. Ricca, B. Simeone, 2008
Polynomial Algorithms for Partitioning a Tree into Single-Center Subtrees to Minimize Flat Service Costs.
Networks, *vol 51 (1) pp 78–89, 2008.*

References

M. Maravalle, B. Simeone, 1995
A spanning tree heuristic for regional clustering.
Communication in Statistics- Theory and Methods. 01/1995; 24(3):625-639.

B. Simeone, 1978
Optimal graph partitioning.
Atti giornate di lavoro AIRO, Urbino. pp 57–73, 1978;

THANK YOU