THE MINIMUM ENERGY BROADCAST PROBLEM I.E. THE MINIMUM SPANNING TREE PROBLEM

Prof. Tiziana Calamoneri

Network Algorithms

A.y. 2022/23

THE PROBLEM

BROADCAST (1)

- Wireless Sensor Networks, once deployed, perform unattended operation for quite a long period.
- During their lifetimes, it is necessary and unavoidable to fix software bugs, reconfigure system parameters, and upgrade the software in order to achieve reliable system performance.
- Especially for a large Wireless Sensor Networks, manually collecting and reconfiguring nodes is infeasible

↓

data dissemination, *i.e.* broadcast

BROADCAST (2)

Broadcast spreads data from a sink node to all nodes in the network, through wireless communication.

Data can be a code image of a renewed program, system commands, or updated system parameters…

There are three requirements of data dissemination in Wireless Sensor Networks:

BROADCAST (3)

- Reliability: all the nodes in the network are covered. Since data dissemination is the building block of many services such as reprogramming and parameter distribution, even a single node not reached may result in inconsistency or crash of the whole network.
- Energy efficiency: the process must be done with minimal energy consumed. This is the consequence of limited power resources.
  The consumed energy consists of read-write and transmission. The read-write is inevitable for storing data blocks. Transmission activity is the major part of energy consumption and also the part that can be controlled.
- Scalability: the number of nodes and the node density may vary. The dissemination protocol is scalable if the completion time of dissemination is linearly increasing with network scale.

THE PROBLEM (1)

- As we already know, a wireless *ad-hoc* network consists of a set *S* of (fixed) radio stations joint by wireless connections.
- We assume that stations are located on the Euclidean plane (only partially realistic hp).
- Nodes have omnidirectional antennas: each transmission is listened by all the neighborhood (natural broadcast).
- …

THE PROBLEM (2)

What does it means "sufficiently close"?…

- Two stations communicate either directly (single-hop) -if they are sufficiently close- or through intermediate nodes (multi-hop).
- A transmission range is assigned to every station: a *range assignment* $r : S \rightarrow R$ determines a directed communication graph $G=(S,E)$, where edge $(i, j) \in E$ iff dist$(i, j) \leq r(i)$ (dist(i, j)= euclidean distance between i and j).
- In other words, $(i, j) \in E$ iff j belongs to the disk centered at i and having radius $r(i)$.

THE PROBLEM (3)

- For reasons connected with energy saving, each station can dynamically modulate its own transmission power.
- In fact, <u>the transmission radius</u> of a station <u>depends on the energy power supplied to the station</u>.

THE PROBLEM (4)

- In particular, the power P_s required by a station *s* to transmit data to another station *t* must satisfy:

$$\frac{P_s}{dist(s,t)^{\alpha}} \geq 1$$

where α≥1 is the distance-power gradient.

Usually $2 \leq \alpha \leq 4$ (it depends on the envoirnment).

In the empty space α=2.

- Hence, in order to have a communication from *s* to *t*, power P_s must be proportional to $dist(s,t)^{\alpha}$.

THE PROBLEM (5)

The general aim is to save energy as much as possible:

- In sensor networks:
  - sensors are battery powered and the consumption of one single battery may imply network failures.
  - main interest: to study the minimisation of the maximum transmission range associated to each sensor
- When devices are more powerful than sensors:
  - all the devices depend on some common electricity generator (*e.g.,* when the deployment of the stations is made in an ad hoc fashion and the electricity to which devices are connected is centralised).
  - main interest: to take care of the total energy consumption

THE PROBLEM (6)

- Stations of an *ad hoc* network cooperate in order to provide specific network connectivity properties by adapting their transmission ranges and, at the same time, they try to save energy.
- …

THE PROBLEM (7)

… According to the required property, different problems are proposed.

For example, the transmission graph:

- is required to be strongly connected

  the problem is NP-hard and there is a 2-approximate alg. in 2 dim. [Kirousis, Kranakis, Krizanc, Pelc '01]; there exists an *r>1* s.t. the problem is not *r*-approximable.

- is required to have diameter at most *h*

  Not trivial; approximate results are not known.

- Given a source node *s*, is required to include a spanning tree rooted at *s* …

THE PROBLEM (8)

In this latter case:

- A Broadcast Range Assignment (for short *Broadcast*) is a range assignment that yields a communication graph *G* containing a directed spanning tree rooted at a given source station *s*.
- A fundamental problem in the design of *ad-hoc* wireless networks is the Minimum-Energy Broadcast problem (for short *Min Broadcast*), that consists in finding a broadcast of minimum *overall energy*.

INAPPROXIMABILITY OF MinBroadcast (1)

Th. *Min Broadcast is not approximable within any constant factor.*

Proof. Recall the *MinSetCover* problem:

given a collection *C* of subsets of a finite universe set *U*, find a subset *C'* of *C* with min cardinality, s.t. each element in *U* belongs to at least one element of *C'*.

Example:

$U=\{1,2,3,4,5\}$ $C=\{\{1,2\}, \{1,2,3\}, \{3\}, \{3,4,5\}\}$

$C'=\{\{1,2,3\},\{3,4,5\}\}$

INAPPROXIMABILITY OF MinBroadcast (2)

Proof (cntd).

Note. *MinSetCover* is not approximable within $c \log n$ for some constant $c>0$, where $n=|U|$.

We will prove that, given an instance *x* of *MinSetCover*, it is possible to construct an instance *y* of *MinBroadcast* s.t. there exists a solution for *x* of cardinality *k* iff there exists a solution for y of cost *k+1*.

So, if *MinBroadcast* is approximable within a constant, then even *MinSetCover* is.

Contradiction.

INAPPROXIMABILITY OF MinBroadcast (3)

Proof (cntd).

Reduction:

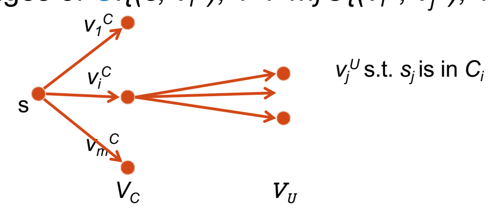
$x=(U,C)$ instance of *MinSetCover* with:

$U=\{s_1, s_2, \ldots, s_n\}$ and $C=\{C_1, C_2, \ldots, C_m\}$.

We construct $y=(G,w,s)$ of *MinBroadcast.*

Nodes of G: $\{s\} \cup \{V_C\} \cup \{V_U\}$

Edges of G: $\{(s, v_i^C), 1 \le i \le m\} \cup \{(v_i^C, v_j^U), 1 \le i \le m, \text{ s.t. } s_j \text{ in } C_i\}$



INAPPROXIMABILITY OF MinBroadcast (4)

Proof (cntd).

Finally, define $w(e)=1$ for any edge e.

Let C' be a solution for x.

A sol. for y assigns 1 to s and to all nodes of V_C in C'.

The resulting transmission graph contains a spanning tree rooted at s because each element in U is contained in at least one element of C'. The cost of such a solution is $|C'|+1$.

INAPPROXIMABILITY OF MinBroadcast (5)

Proof (cntd).

…

Conversely, assume that r is a feasible sol. for y, (w.l.o.g. $r(v)$ is either 0 or 1 if v is in V_C: other values would be meaningless) and $r(v)=0$ if v is in V_S.

We derive a solution C' for x selecting all subsets C_i s.t. $r(v_i^C)=1$.

It holds that $|C'|=cost(r)-1$. ■

EUCLIDEAN MinBroadcast (1)

Note:

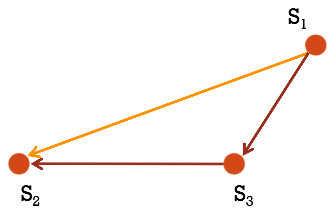
We proved that *Min Broadcast* is not approximable within a constant factor, but we have dealt with the general problem.

There are some special cases (*e.g.* the <u>Euclidean bidimensional</u> one) that are particularly interesting and that behave better!

In the following, we restrict to the special case of Euclidean plane…

EUCLIDEAN MinBroadcast (2)

- **Obs.** Collaborating in order to minimize the overall energy is crucial:



  - S_1 needs to communicate with S_2
  - let $\alpha=2$
  - Cost of $S_1 \rightarrow S_2 = dist(S_1, S_2)^2$
  - Cost of $S_1 \rightarrow S_3 \rightarrow S_2 = dist(S_1, S_3)^2 + dist(S_3, S_2)^2$
  - When angle $S_1S_3S_2$ is obtuse: $dist(S_1, S_2)^2 > dist(S_1, S_3)^2 + dist(S_3, S_2)^2$

EUCLIDEAN MinBroadcast (3)

- In the Euclidean case, a range assignment r can be represented by the correspondent family $D = \{D_1, \ldots, D_l\}$ of disks, and the overall energy is defined as:

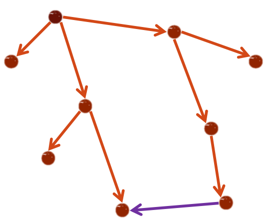
$$\cos t(D) = \sum_{i=1}^{l} r_i^{\alpha}$$

where r_i is the radius of D_i.

EUCLIDEAN MinBroadcast (4)

- Consider the complete and weighted graph $G^{(\alpha)}$ where the weight of each arc $e=(u,v)$ is $dist(u,v)^{\alpha}$.
- The broadcast problem is strictly related with the minimum spanning tree on $G^{(\alpha)}$, in view of some important properties…

EUCLIDEAN MinBroadcast (5)



The unavoidable set of connections used to perform a broadcast from s:

- cannot generate a cycle, because nodes do not need to be informed twice

⬇

tree

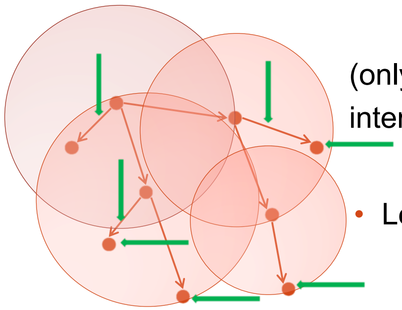
- minimizes the overall energy

⬇

long arcs waste more energy than short ones.

EUCLIDEAN MinBroadcast (6)

- Nevertheless, the Minimum Broadcast problem is not the same as the Min Spanning Tree problem:



- The energy used by each node u is $\max_{(u,v)\in T}\{dist(u,v)\}^{\alpha}$ (only the longest edge of each internal node gives its contribution)
- Leaves waste no energy

EUCLIDEAN MinBroadcast (7)

- The Minimum Broadcast problem is NP-hard in its general version and it is neither approximable within (1-ε)*Δ,* where *Δ* is the maximum degree of *T* and ε is an arbitrary constant.
- Nothing is known about the hardness of the geometric version (*i.e.* in the Euclidean plane).

EUCLIDEAN MinBroadcast (8)

- An approximation algorithm is based on the computation of the MST:
  - compute the MST of the complete graph induced by *U*,
  - Assign a direction to arcs (from *s* to the leaves)
  - Assign to each node *i* a radius equal to the length of the longest arc outgoing from *i*
- Easy to implement ➔ deep analysis of the approx ratio.
  - [Clementi+al.'01] the first constant approx ratio (about 40)
  - [Ambüehl '05] the best (tight) known approx ratio (6)

THE MINIMUM SPANNING TREE PROBLEM (RECUP)

MINIMUM SPANNING TREE (1)

- **Obs. 1**: If the weights are *positive*, then a MST is in fact a minimum-cost subgraph connecting all nodes.
- Proof: A subgraph containing cycles necessarily has a higher total weight. ■

- **Obs. 2**: There may be several minimum spanning trees of the same weight having a minimum number of edges.
- In particular, if all the edge weights of a given graph are the same, then every spanning tree of that graph is minimum.

MINIMUM SPANNING TREE (2)

- **Obs. 3**: If each edge has a distinct weight, then there is a unique MST.
- This is true in many realistic situations, where it's unlikely that any two connections have *exactly* the same cost
- Proof: Assume by contradiction that MST T is not unique. So, there is another MST with equal weight, say T'.

…

MINIMUM SPANNING TREE (3)

(proof – cntd)

- Let e_1 be an edge that is in T but not in T'. As T' is a MST, $\{e_1\} \cup T'$ contains a cycle C and there is at least one edge e_2 in T' that is not in T and lies on C.
- If the weight of e_1 is less than that of e_2:

  replacing e_2 with e_1 in T' yields tree $\{e_1\} \cup T' \setminus \{e_2\}$ which has a smaller weight compared to T'.

  Contradiction, as we assumed T' is a MST but it is not.
- If the weight of e_1 is larger than that of e_2:
  a similar argument involving tree $\{e_2\} \cup T \setminus \{e_1\}$ also leads to a contradiction.
- We conclude that the assumption that there is a further MST was false. ■

MINIMUM SPANNING TREE (4)

- **Obs. 4**: For any cycle C in the graph, if the weight of an edge e of C is larger than the weights of all other edges of C, then this edge cannot belong to an MST.
- Proof: Assuming the contrary, i.e. that e belongs to an MST T_1, then deleting e will break T_1 into two subtrees with the two endpoints of e in different subtrees. The remainder of C reconnects the subtrees, in particular there is an edge f of C with endpoints in different subtrees, i.e., it reconnects the subtrees into a tree T_2 with weight less than that of T_1, because the weight of f is less than the weight of e. ■

MINIMUM SPANNING TREE (5)

- **Obs. 5**: If in a graph there exists a unique edge e with the minimum weight, then this edge is included in any MST.
- Proof: If e was not included in the MST, removing any of the (larger cost) edges in the cycle formed after adding e to the MST, would yield a spanning tree of smaller weight. ■

MINIMUM SPANNING TREE (6)

- **Obs. 6**: For any cut C in the graph, if there exists a unique edge e of C with minimum weight in C, then this edge is included in any MST.
- Proof: If e was not included in the MST, adding e to the MST produces a cycle. Removing any of the (larger cost) edges of the cut in the cycle, would yield a spanning tree of smaller weight. ■
- By similar arguments, if more than one edge is of minimum weight across a cut, then each such edge is contained in a minimum spanning tree.

MINIMUM SPANNING TREE (7)

Three classical algorithms:

- Kruskal ['56]
- Prim ['57]
- Boruvka ['26]

MINIMUM SPANNING TREE (8)

- The three algorithms are all greedy algorithms and based on the same structure:
  - Given a set of arcs A containing some MST arcs, e is a safe arc w.r.t. A if $A \cup \{e\}$ contains only MST arcs, too.
  - A=empty set

    While A is not a MST

    find a safe arc e w.r.t. A “difficult” issue

    $A=A \cup \{e\}$

MINIMUM SPANNING TREE (9)

- A=empty set
  while A is not a MST
  find a safe arc e w.r.t. A
  $A=A \cup \{e\}$

where:

- A is acyclic
- graph $G_A=(V, A)$ is a forest whose each connected component is either a node or a tree
- Each safe arc connects different connected components of G_A
- the while loop is run $n-1$ times

KRUSKAL ALGORITHM (1)

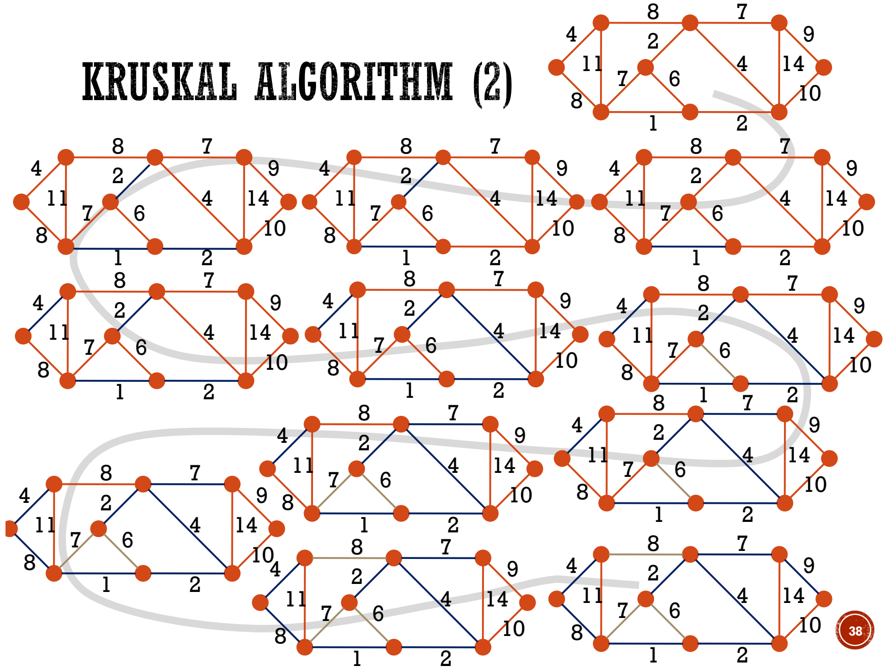
- *A*=empty set
  While G_A is not a MST
  find a safe arc *e* w.r.t. *A*
  A=*A* U {*e*}

Among those connecting two different connected components in G_A, choose the one with minimum weight

Implementation using:

- Data structure Union-Find
- The set of the arcs of G_A is sorted w.r.t. their weight
- Time Complexity: *O(m log n)* [Johnson '75, Cheriton & Tarjan '76]

KRUSKAL ALGORITHM (2)



PRIM ALGORITHM (1)

- *A*=empty set
  While G_A is not a MST
  find a safe arc *e* w.r.t. *A*
  A=*A* U {*e*}

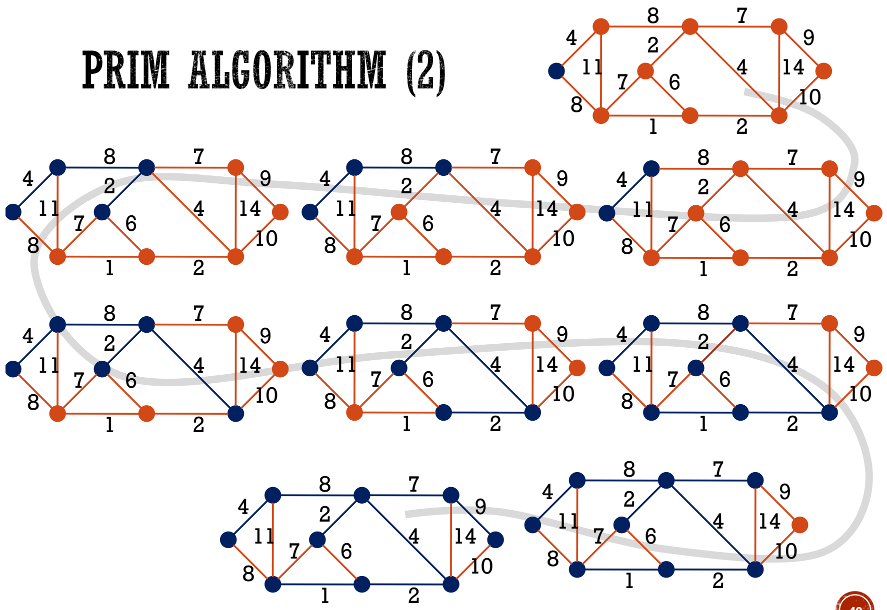
Among those connecting the main connected component with an isolated node, choose the one with minimum weight

Implementation using:

- Nodes in a min-priority queue w.r.t. *key(v)*=min weight of an arc connecting *v* to a node of the main connected component; ∞ if it does not exist
- Priority queue = heap ➔ Complexity: *O(m log n)*
- Priority queue = Fibonacci heap ➔ Complexity: *O(m+n log n)*

[Ahuja, Magnanti & Orlin '93]

PRIM ALGORITHM (2)



BORUVKA ALGORITHM (1)

(purpose: an efficient electrical coverage of Mor

Among those connecting C_i to another component, the one with minimum weight

Hypothesis: each arc has a distinct weight

- *A*=empty set

  While *A* is not a MST

    for each connected component C_i of G_A

      find a safe arc e_i w.r.t. C_i

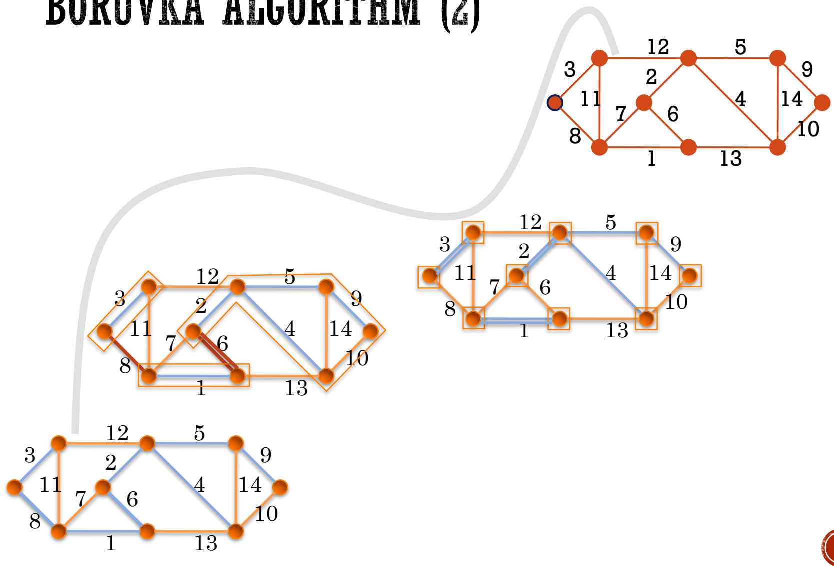
      $A = A \cup \{e_i\}$

Trick: handle many arcs (exactly log of the # of connected components) during the same loop

Impossible to introduce cycles, thanks to the hipothesis!

Complexity: *O(m log n)*

BORUVKA ALGORITHM (2)



OTHER ALGORITHMS (1)

- [Friedman & Willard '94] Linear time algorithm, but it assumes the edges are already sorted w.r.t. their weight. Not used in practice, as the asymptotic notation hides a huge constant.
- [Matsui '95] Linear time algorithm for planar graphs (possible lesson)

OTHER ALGORITHMS (2)

- [Frederickson '85, Eppstein '94] Given a graph and its MST, it is even interesting to find a new MST after that the original graph has been slightly modified. It can be performed in average time *O(log n)*
- Only *O(n+m)* time is necessary to verify whether a given spanning tree is minimum.

AGAIN ON MINIMUM ENERGY BROADCAST

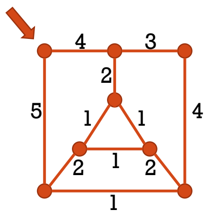
HEURISTICS (1)

In [Wieselthier, Nguyen, Ephremides, 00]: three heuristics all based on the greedy technique:

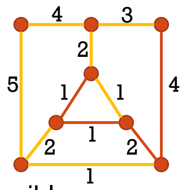
- SPT (spanning path tree): it runs Dijkstra algorithm to get the minimum path tree, then it directs the edges of the tree from the root to the leaves.
- BAIP (Broadcast Average Incremental Power): it is a modification of the Dijkstra algorithm based on the nodes (i.e. a new node is added to the tree on the basis of its minimum average cost).
- MST (min spanning tree): it runs Prim algorithm to get a MST, then it directs the edges of the tree from the root to the leaves.

A PARENTHESIS

SPT (spanning path tree) and MST (min spanning tree) can be different:



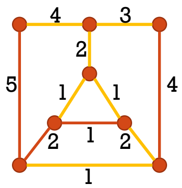
Dijkstra algorithm (SPT)



Possibly many trees, not all of the same weight

(e.g., add the upper edge of weight 1)

Prim algorithm (MST)



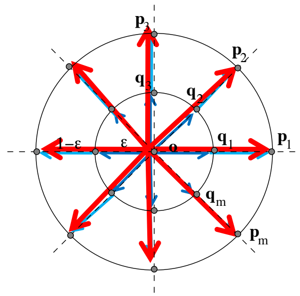
Possibly many trees, all of the same (minimum) weight

HEURISTICS (2)

GREEDY IS NOT ALWAYS GOOD

Greedy is not always good [Wan, Calinescu, Li, Frieder '02]:

- SPT: it runs Dijkstra algorithm to get the minimum path tree, then it directs the edges of the tree from the root to the leaves



(let $\alpha=2$)

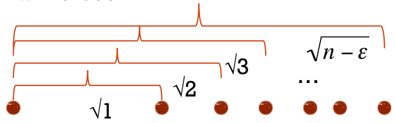
- SPT outputs a tree with total energy:

  $\varepsilon^2+n/2(1-\varepsilon)^2$
- If the root transmits with radius 1 the energy is 1
- When $\varepsilon \to 0$ SPT is far $n/2$ from the optimal solution.

HEURISTICS (3)
GREEDY IS NOT ALWAYS GOOD

- BAIP (Broadcast Average Incremental Power): it is a modification of the Dijkstra algorithm based on the nodes: a new node is added to the tree on the basis of the min average cost = energy increasing / # of added nodes.
- It has been designed to solve the problems of SPT.

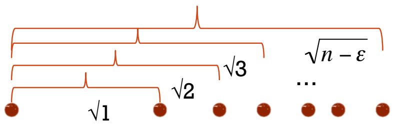
HEURISTICS (4)
GREEDY IS NOT ALWAYS GOOD



(let α=2):

- The min transmission power of the source to reach k receiving nodes is $\sqrt{k^2} = k$ and thus the average power efficiency is $k / k = 1$
- On the other hand, the min transmission power of the source to reach all receiving nodes is $(\sqrt{n-\varepsilon})^2 = n-\varepsilon$ and thus the average power efficiency is $(n-\varepsilon)/n=1 - \varepsilon/n$…

HEURISTICS (5)
GREEDY IS NOT ALWAYS GOOD



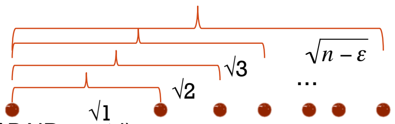
- BAIP will let the source to transmit at power $\sqrt{n-\varepsilon}$ to reach all nodes in a single step.
- However, the opt. routing is a path consisting of all nodes from left to right. Its min power is:

$$\sum_{i=1}^{n-1}(\sqrt{i}-\sqrt{i-1})^2+(\sqrt{n-\varepsilon}-\sqrt{n-1})^2<\sum_{i=1}^{n}(\sqrt{i}-\sqrt{i-1})^2=$$

$$\sum_{i=1}^{n}(\sqrt{i}-\sqrt{i-1})^2\frac{(\sqrt{i}+\sqrt{i-1})^2}{(\sqrt{i}+\sqrt{i-1})^2}=\sum_{i=1}^{n}\frac{((\sqrt{i}-\sqrt{i-1})(\sqrt{i}+\sqrt{i-1}))^2}{(\sqrt{i}+\sqrt{i-1})^2}=$$

$$=\sum_{i=1}^{n}\frac{(i-(i-1))^2}{(\sqrt{i}+\sqrt{i-1})^2}=\sum_{i=1}^{n}\frac{1}{(\sqrt{i}+\sqrt{i-1})^2}=1+\sum_{i=2}^{n}\frac{1}{(\sqrt{i}+\sqrt{i-1})^2}\le$$

HEURISTICS (6)
GREEDY IS NOT ALWAYS GOOD



(computation of the performance ratio of BAIP – cntd)

$$\le 1+\sum_{i=2}^{n}\frac{1}{i+(i-1)+2\sqrt{i}\sqrt{i-1}}\le 1+\sum_{i=2}^{n}\frac{1}{2i-1+2(i-1)}\le$$

$$\le 1+\sum_{i=2}^{n}\frac{1}{2i-1+2(i-1)}=1+\sum_{i=2}^{n}\frac{1}{4i-3}\le 1+\sum_{i=2}^{n}\frac{1}{4(i-1)}\le$$

Substituting $i=j+1$:

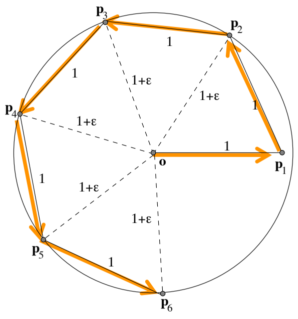
$$\le 1+\sum_{j=1}^{n-1}\frac{1}{4j}\le 1+\frac{1}{4}\sum_{j=1}^{n-1}\frac{1}{j}\le 1+\frac{1}{4}(\ln(n-1)+1)=\frac{\ln(n-1)+5}{4}$$

Thus the approx ratio of BAIP is at least:

$$\frac{n-\varepsilon}{\frac{\ln(n-1)+5}{4}}\rightarrow(\varepsilon\rightarrow 0)\frac{4n}{\ln(n-1)+5}=\frac{4n}{\ln n}+o(1)$$

HEURISTICS (7)
GREEDY IS NOT ALWAYS GOOD

MST: it runs Prim algorithm to get a MST, then it directs the edges of the tree from the root to the leaves



- Path $op_1 \ldots p_6$ is the unique MST, and its total energy is 6.
- On the other hand, the opt. routing is the star centered at *o*, whose energy is $(1+\varepsilon)^{\alpha}$.
- The approx. ratio converges to 6, as ε goes to 0.

HEURISTICS (8)

- We have just shown a lower bound on the approximation ratio of MST.
- This ratio is a constant and an upper bound is 12.
- The proof involves complicated geometric arguments…

HEURISTICS (13)

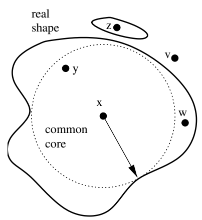
- **Obs.** The proof in [Wan, Calinescu, Li, Frieder '02] contains a small flaw that can be solved, arriving to an approximation ratio of 12,15 [Klasing, Navarra, Papadopoulos, Perennes '04]
- Indipendently, an approximation ratio of 20 has been stated in [Clementi, Crescenzi, Penna, Rossi, Vocca '01]
- Approx. ratio improved to 7,6 [Flammini, Klasing, Navarra, Perennes '04]
- Approx. ratio improved to 6,33 [Navarra '05]
- Optimal bound 6 [Ambüehl '05]
- For realistic instances, experiments suggest that the tight approximation ratio is not 6 but 4 [Flammini, Navarra, Perennes '06]

HEURISTICS (14)

The 3-dimensional space better models practical environments:

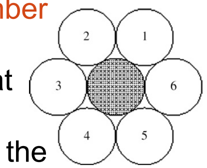
- in real life scenarios, radio stations are distributed over a 3-dimensional Euclidean space.
- the extension to the 3-dimensional case of the assumption that transmissions are propagated uniformly in a spherical shape naturally comes from the 2- dimensional although it is not realistic: in general, in real world scenarios, the propagation is not uniform but a common core (not necessarily connected) covering a sphere:

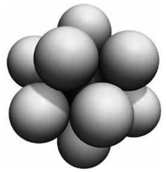
The mismatching between real world scenarios and the uniform assumption might be overcome then by considering higher values for the constant α.



HEURISTICS (15)

- the approximation ratio of 6 for the MST heuristic in the 2-dimensional Euclidean space for α=2 coincides with the 2-dimensional kissing number.
- the *d*-dimensional kissing number is the maximum number of *d*-spheres of a given radius *r* that can simultaneously touch a *d*-sphere of the same radius *r* in the *d*-dimensional Euclidean space





HEURISTICS (16)

- In general, the *d*-dimensional kissing number was proven to be a lower bound for the approximation ratio of the MST heuristic for any dimension $d > 1$ and power $\alpha \geq d$
- The 3-dimensional kissing number is 12, but the best known approximation ratio of the MST heuristic so far is 18,8 [Navarra '08]

☞ student lesson