THE DATA COLLECTION IN SENSOR NETWORKS I.E. THE CONNECTED DOMINATING SET PROBLEM

1

Prof. Tiziana Calamoneri

Network Algorithms

A.y. 2025/26

THE PROBLEM

2

THE DATA COLLECTION PROBLEM (1)

Whenever a data mule is not used, sensing devices run with very low energy consumption to sense and monitor the surrounding environment, so data collection is the main reason of energy consumption.

The aim of the data collection problem is to transfer all of the periodically sensed data to the sink efficiently by one or more hops so that the network lifetime is maximized.

THE DATA COLLECTION PROBLEM (2)

Many approaches to the problem:

- **Naive approach**: each sensor node increases its transmission range to send data directly to the sink, resulting in enormous energy consumption which reduces network lifetime.
- **Multi-hop data routing**: sensor nodes send gathered data to the nearest node on the shortest path to the sink. So, the nodes close to the sink send a large number of data to it and die out quickly, causing an uneven load distribution in network.
- …

THE DATA COLLECTION PROBLEM (3)

Many approaches to the problem (contd)

- **Clusters**: sensors nodes are subdivided into clusters transmitting aggregated data to the sink. Usually, cluster head nodes accumulate data packets from their member nodes and transmit these data to the sink. Since energy dissipation is directly proportional to the traversed distance, an issue is to minimize the inter-cluster and intra-cluster distances.
- …

THE DATA COLLECTION PROBLEM (4)

Many approaches to the problem (contd):

- **Duty cycle based mode**: sensor transmitters are active only in some periods. Sleeping sensors are activated either at fixed intervals or when they sense an event or data. When a sender transmits data (which may be either from a relay node or generated by itself), it establishes a link for data transmission if there are active nodes among its neighbors. If all of them are in a sleep state, the sender needs to remain active until someone wakes up and data can be transmitted. So, this mode increases the transmission delay of the network.

A POSSIBLE MIXED APPROACH (1)

Approach from [Sal19] based on the duty cycle mode:

A part of the nodes is selected to construct a connected sub-network (backbone), whose nodes adopt a periodic sleep/awake working mode at fixed intervals, while the other nodes turn off the radio device when there is no data to transmit, and only sensing surrounding environment.
When there is data to be sent, the radio device is turned on and send the data to the nodes in the backbone, then route the data to the sink through the nodes in the backbone.

A POSSIBLE MIXED APPROACH (2)

Approach based on the duty cycle mode (contd)

Thus, many nodes are in the sleep state most of the time, which can save a lot of energy.
Instead, the energy consumption of the nodes in the backbone is relatively high.
Therefore, after the backbone works for a certain period of time, the nodes with more residual energy are selected to reconstruct a *new* backbone, so that the energy consumption of nodes in the network is more balanced, which improves the network lifetime.

A POSSIBLE MIXED APPROACH (3)

Approach based on the duty cycle mode (contd)

Requirements of the backbone:

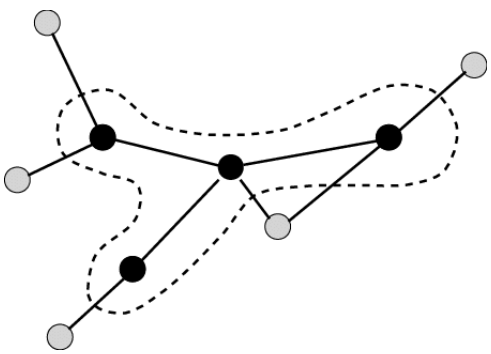
(a) # of nodes in the backbone as small as possible (**min**);

(b) at least one route to the sink for each node in the backbone (**connected**);

(c) The other nodes in the network must communicate directly with at least one node in the backbone (**dominating set**).

In other words, these nodes constitute a min connected dominating set…

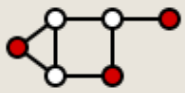
MIN CONNECTED DOMINATING SET



MIN CONNECTED DOMINATING SET (1)

Def. A dominating set (DS) for a graph G = (V, E) is a subset D of V such that every node not in D is adjacent to at least one member of D (e.g., fig (a))

(a)



A min DS is a dominating set with smallest possible cardinality, call it d (e.g., fig (b)).

Def. A (min) connected dominating set (CDS) for a graph G = (V, E) is a subset D of V such that:

- D induces a connected subgraph of G
- D is a (min) DS (e.g., fig. (c)).

MIN CONNECTED DOMINATING SET (2)

Let d be the cardinality of a min CDS.

Any spanning tree T of G has at least two leaves.

A maximum leaf spanning tree is a spanning tree that has the largest possible number of leaves among all spanning trees of G. Call it l.

Theorem. In any n-node graph G, where $n > 2$, $n = d + l$.

- Proof. …

MIN CONNECTED DOMINATING SET (2)

Proof of Thm: n=d+l.

Proof. Prove the two inequalities:

- If D is a CDS, then there exists a spanning tree for G whose leaves include all nodes that are not in D : start from the (connected) subgraph induced by D and add as leaf each node v that is not in D to a neighbor of v in D. Let $l' \leq l$ its # of leaves. Then, by construction, $l \geq l' = n - d$.
- Let T be the max leaf spanning tree of G. The nodes of T that are not leaves form a CDS D' of G, so |D'|=d' ≥d. This shows that $n - l = d' \geq d$.
- Putting these two inequalities together, we have $n=d+l$.

COMPUTATIONAL COMPLEXITY (1)

Computationally, this implies that determining the connected domination number is equally difficult as finding the max leaf number.

It is NP-complete to test whether there exists a CDS with size less than a given threshold, or equivalently to test whether there exists a spanning tree with at least a given number of leaves.

COMPUTATIONAL COMPLEXITY (2)

- In terms of approximation algorithms, connected domination and maximum leaf spanning trees are not the same: there exists an approximation for the min CDS that achieves a factor of $2 \ln \Delta + O(1)$, where Δ is the maximum degree of G [Guha&Khuller'98] while the max leaf spanning tree problem can be approximated within a factor of 2 [Solis-Oba&Roberto'98].
- In graphs of maximum degree 3, the CDS and its complementary maximum leaf spanning tree problem can be solved in polynomial time [Ueno et al.'88].

MIN DOMINATING SET (1)

If we drop the constraint to induce a connected subgraph, we require to find a min dominating set, studied from the 1950s onwards, whose rate of research significantly increased in the mid-1970s.

- …

A PARENTHESIS ON MIN SET COVER

Set Cover Problem:

Given a set of subsets $S=\{S_1, \ldots, S_n\}$ of the universal set U such that

$$\bigcup_{i=1..m} S_i = U$$

what is the smallest subset J of {1,…,n} such that

$$\bigcup_{i \in J} S_i = U \ ?$$

In 1972, Karp proved the set cover problem to be NP-complete, with immediate implications for the dominating set problem:

- …

MIN DOMINATING SET (2)

Theorem. There is a bijection between the solutions of min dominating set and min set cover problems.

Proof. The following two reductions show that the minimum dominating set problem and the set cover problem are equivalent under L-reductions: given an instance of one problem, we can construct an equivalent instance of the other problem.

…

MIN DOMINATING SET (3)

Bijection between min dominating set and min set cover (contd)

From dominating set to set cover

Given $G = (V, E)$ with $V = \{1, 2, \ldots, n\}$, construct a set cover instance (U, S) as follows:

- the universe U is V,
- the family of subsets is $S = \{S_1, S_2, \ldots, S_n\}$ such that S_v consists of node v and all its adjacent nodes.

Now, if D is a dominating set for G, then $C = \{S_v : v \in D\}$ is a feasible set cover, with $|C| = |D|$.

Conversely, if $C = \{S_v : v \in D\}$ is a set cover, then D is a dominating set for G, with $|D| = |C|$.

...

MIN DOMINATING SET (4)

Bijection between min dominating set and min set cover (cntd)

Example: Given G

construct a set cover instance with universe $U = \{1, 2, 3, 4, 5, 6\}$ and subsets:

$S_1 = \{1, 2, 5\}$, $S_2 = \{1, 2, 3, 5\}$, $S_3 = \{2, 3, 4, 6\}$,

$S_4 = \{3, 4\}$, $S_5 = \{1, 2, 5, 6\}$, and $S_6 = \{3, 5, 6\}$.

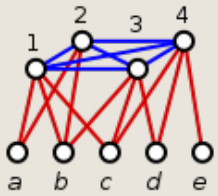
$D = \{3, 5\}$ is a dominating set for G and corresponds to the set cover $C = \{S_3, S_5\}$.

MIN DOMINATING SET (5)

Bijection between min dominating set and min set cover (cntd)

From set cover to dominating set

Let (U,S) be an instance of set cover with universe U and the family of subsets $S = \{S_i : i \in I\}$; assume that U and the index set I are disjoint.



Construct graph G = (V, E) as follows:

- the set of nodes is $V = I \cup U$,
- there is an edge $\{i, j\} \in E$ between each pair $i, j \in I$, and there is also an edge $\{i, u\}$ for each $i \in I$ and $u \in S_i$. It turns out that G is a split graph: I is a clique and U is an independent set.

MIN DOMINATING SET (6)

Bijection between min dominating set and min set cover (cntd)

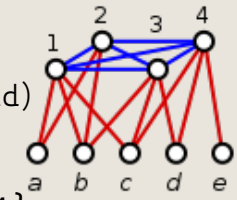
Now if $C = \{S_i : i \in D\}$ is a set cover for some $D \subseteq I$, then D is a dominating set for G, with $|D| = |C|$.

Indeed: for each $u \in U$ there is an $i \in D$ such that $u \in S_i$, and by construction, u and i are adjacent in G; hence u is dominated by i; moreover, since D must be nonempty, each $i \in I$ is adjacent to a node in D.

Conversely, let D be a dominating set for G. Then construct another dominating set X s.t. $|X| \leq |D|$ and $X \subseteq I$: simply replace each $u \in D \cap U$ by a neighbour $i \in I$ of u. Then $C = \{S_i : i \in X\}$ is a set cover, with $|C| = |X| \leq |D|$.

MIN DOMINATING SET (7)

Bijection between min dominating set and min set cover (cntd)



Example Here U = {a, b, c, d, e}, I = {1, 2, 3, 4},

S_1 = {a, b, c}, S_2 = {a, b}, S_3 = {b, c, d}, S_4 = {c, d, e}.

Let C = {S_1, S_4} be a set cover; this corresponds to the dominating set D = {1, 4}.

D = {a, 3, 4} is another dominating set for G. Given D, we can construct a dominating set X = {1, 3, 4} which is not larger than D and is a subset of I.

Dominating set X corresponds to set cover C={S_1, S_3, S_4}.

MIN DOMINATING SET (8)

In view of this equivalence, not only the dominating set problem is NP-complete as well, but an efficient algorithm for min dominating set would provide an efficient algorithm for set cover, and vice-versa.

Moreover, the reductions preserve the approximation ratio: for any α, a polynomial α-approx algorithm for min dominating set would provide a polynomial α-approx algorithm for set cover and vice-versa.

GREEDY ALGORITHMS FOR MIN CDS

A TWO-STEP GREEDY ALGORITHM (1)

Two-step greedy algoritm [Guha & Khuller '98]:

- Consider G and a subset C of its nodes.
- All nodes in G can be divided into three classes w.r.t. C:
  - B (Black): nodes in C
  - Gr (Gray): nodes not in C but adjacent to C
  - W (White): nodes not in C and not adjacent to C

Clearly, B *U* Gr U W=V, and C is a CDS if and only if there is no white node AND the subgraph induced by black nodes is connected.

CC: # of conn. components in this black subgraph.

Then: |W|+CC=1.

A TWO-STEP GREEDY ALGORITHM (2)

Greedy algorithm based on the potential function $Pf = |W| + CC$:

First-Step Greedy Algorithm (G)

Repeat

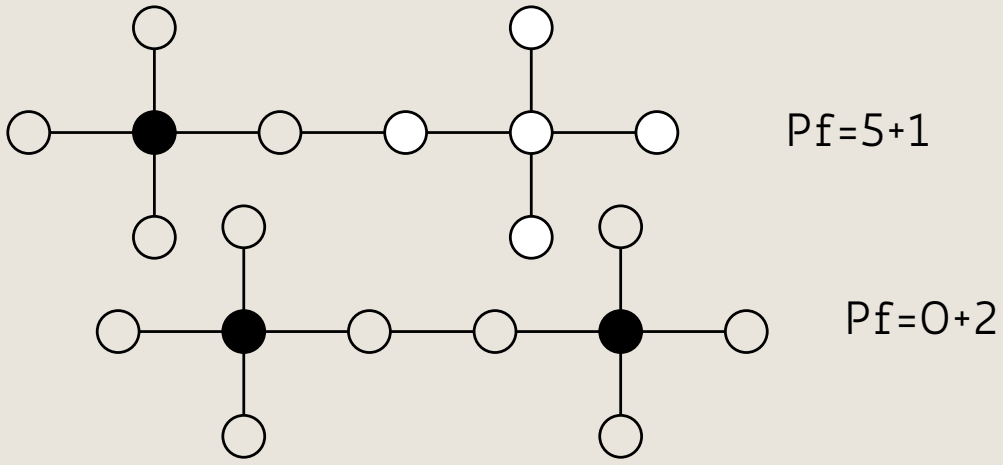
if there exists a white or gray node s. t. coloring it in black and its adjacent white nodes in gray would reduce the value of Pf

then choose such a node and reduce the value of Pf

else return

A TWO-STEP GREEDY ALGORITHM (3)

Clearly, when the loop ends, no white node will exist, i.e., all black nodes form a dominating set:



however, the subgraph induced by black nodes may be not connected…

A TWO-STEP GREEDY ALGORITHM (4)

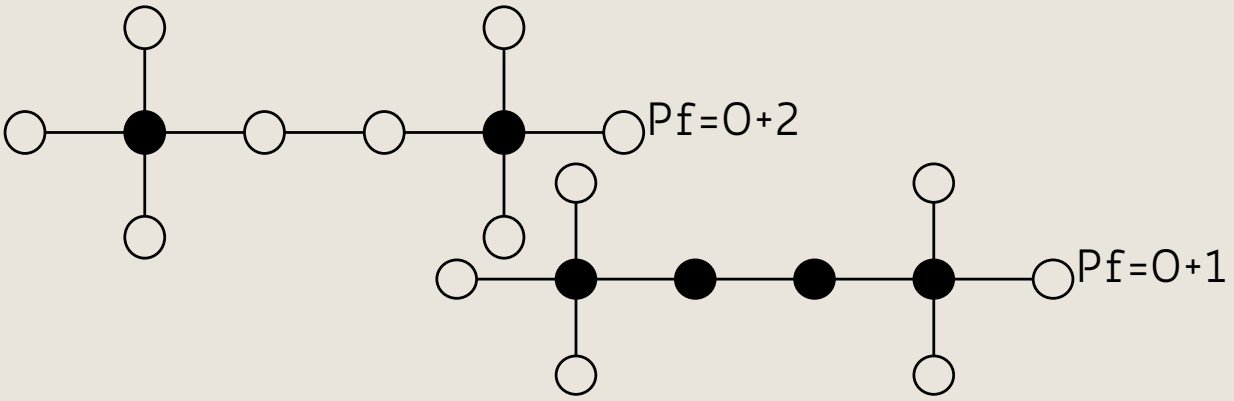
Second-Step Greedy Algorithm (G)

Repeat

color either one or two gray nodes in black to reduce CC

Until CC=1

This step guarantees to obtain a connected dominating set.

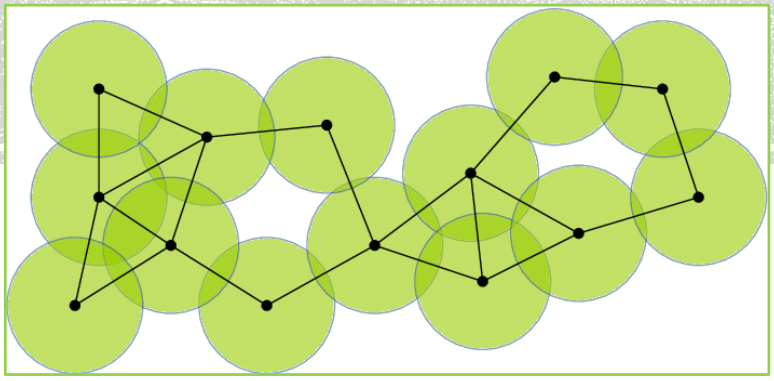


A ONE-STEP GREEDY ALGORITHM

The authors prove that the approximation ratio is 3+ln D with D = max degree of G.

It is possible to introduce a more complex potential function, design a single-step greedy algorithm and get a better approximation ratio of 2+ln D [Ruan et al. '04]

MIN CDS IN UNIT DISK GRAPHS

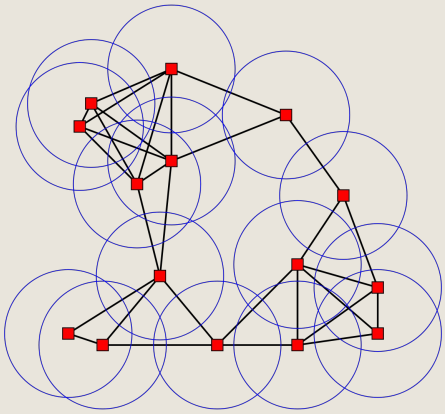


31

UNIT DISK GRAPHS (1)

Consider a set of n equal-sized circles in the plane.

The intersection graph of these circles is an n-node graph; each node corresponds to a circle, and there is an edge between two nodes when the corresponding circles intersect (tangent circles are assumed to intersect).

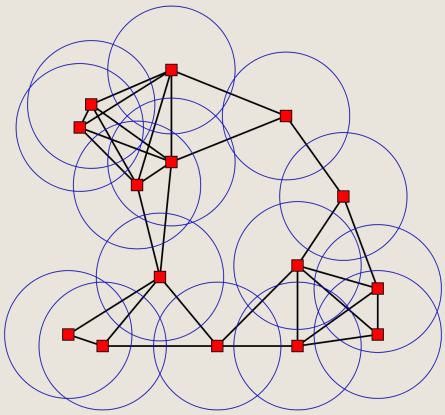


32

UNIT DISK GRAPHS (2)

Such intersection graphs are called unit disk graphs, and the set of n circles is an intersection model.

Disk graphs are suitable to model wireless networks: each circle center is a transceiver and the radius represent the transmission radius. If the network is homogeneous, the circles will be approximately equal in size and, w.l.o.g., it is equal to 1.



MIN CDS IN UNIT DISK GRAPHS (1)

When restricted to UDGs, min CDS is still NP-hard [Lichtenstein '82]. It remains NP-hard when restricted to grids, that are a subclass of UDGs [Clark, Colbourn, Johnson '90].

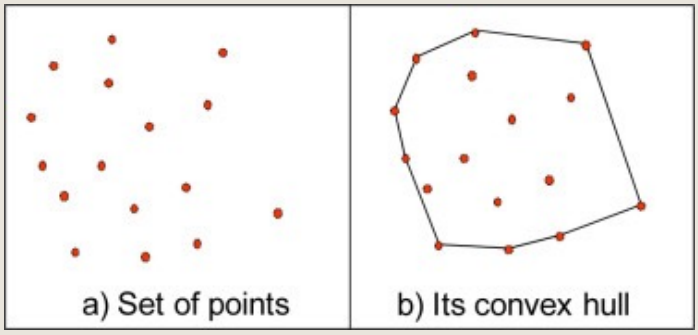
[Cheng et al 2003] gives a polynomial time approximation scheme (PTAS), that is, for any (arbitrarily small) $\varepsilon > 0$, there exists a polynomial-time $(1 + \varepsilon)$-approximation; note that the time is polynomial in the problem size for every fixed ε, but can be different for different ε.

But this algorithm is not used in practice...

MIN CDS IN UNIT DISK GRAPHS (2)

[Purohit & Sharma '10] gives an easy distributed algorithm for UDGs reducing a given (also trivial) CDS:

Def. The convex hull for a set of points X in the 2D space is the minimum convex set containing X.



MIN CDS IN UNIT DISK GRAPHS (3)

Algorithm Distributed_Reduce_CDS

Repeat

Select a min degree node u from the given CDS

Compute CH(N[u]) and, ∀ i ∈ N(u), CH(N[i])

if $CH(N[u]) \subseteq U_{i \in N(u)} CH(N[i])$

then remove node u from the given CDS

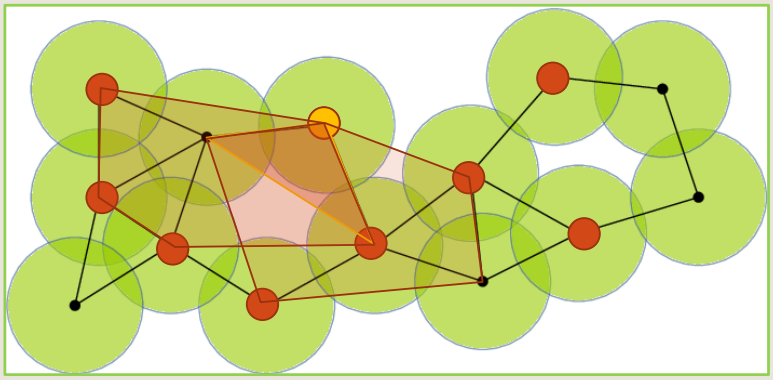
Until there are unconsidered nodes in the given CDS

Note. N[v] is the closed neighborhood, i.e., v and its adjacent nodes.

MIN CDS IN UNIT DISK GRAPHS (4)

Example:

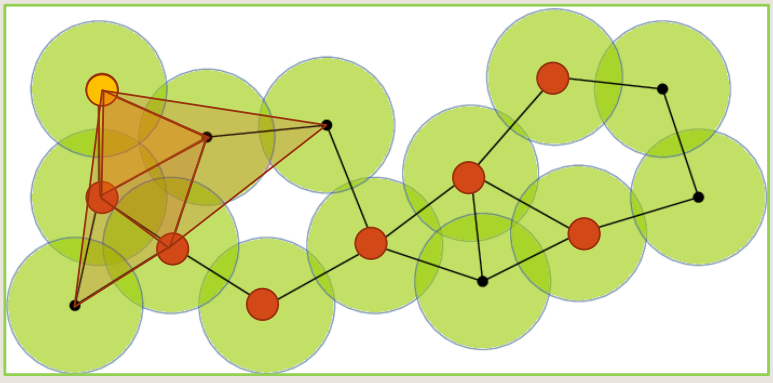
Algorithm Distributed_Reduce_CDS
Repeat
Select a minimum degree node u from the given CDS
Compute CH(N[u]) and, ∀ i ∈ N(u), CH(N[i])
if $CH(N[u]) \subseteq U_{i \in N(u)} CH(N[i])$
then remove node u from the given CDS
Until there are unconsidered nodes in the given CDS



MIN CDS IN UNIT DISK GRAPHS (4)

Example:

Algorithm Distributed_Reduce_CDS
Repeat
Select a minimum degree node u from the given CDS
Compute CH(N[u]) and, ∀ i ∈ N(u), CH(N[i])
if $CH(N[u]) \subseteq U_{i \in N(u)} CH(N[i])$
then remove node u from the given CDS
Until there are unconsidered nodes in the given CDS



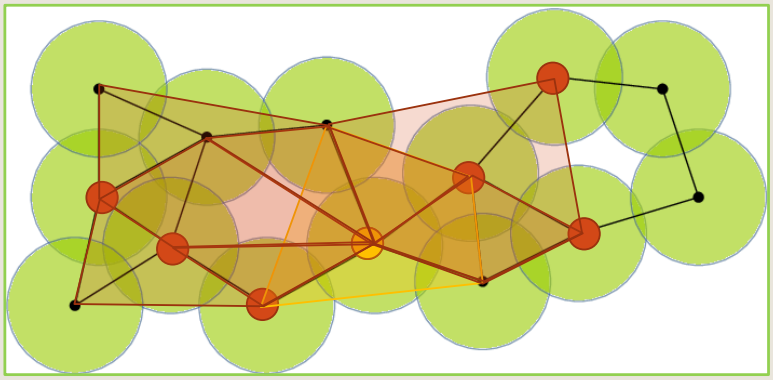
and so on…

MIN CDS IN UNIT DISK GRAPHS (4)

Example:

nodes with larger degree are not worthy to be considered …

Algorithm Distributed_Reduce_CDS
Repeat
Select a minimum degree node u from the given CDS
Compute CH(N[u]) and, ∀ i ∈ N(u), CH(N[i])
if CH(N[u]) ⊆ $U_{i \in N(u)}$ CH (N[i])
then remove node u from the given CDS
Until there are unconsidered nodes in the given CDS



MIN CDS IN UNIT DISK GRAPHS (4)

This algorithm:

- is very easy and sometimes reduces the dimension of the given CDS because it exploits geometric reasonings
- has the merit to work in a distributed fashion (no global knowledge is necessary)

BUT

no approximation ratio is guaranteed!!