Lifetime Maximization of Sensor Networks for Area Monitoring (work in progress)

Dennis Schieferdecker (schiefer@ira.uka.de)
Sanjeev Arora, Peter Sanders, David Steurer

ITI Sanders, Universtiy of Karlsruhe (TH)

23.10.2008

Table of Contents

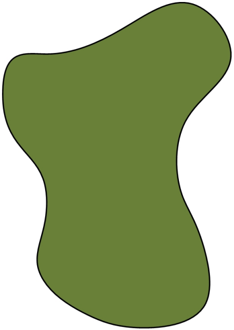
– I –
Area Monitoring
energy-efficient, sensor-based

Overview

Problem Formulation - 1

Area Monitoring

- Permanent monitoring of area F (e.g. temperature profiles, intrusion detection, ...)
- Spreading of N sensor nodes
  - ↪ More sensors than necessary for full coverage of F
- At each point in time, activate only as many sensors as necessary
  - ↪ maximize lifetime T

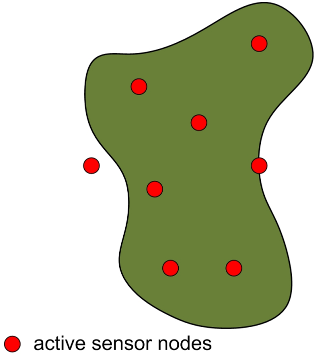


Overview

Problem Formulation - 1

Area Monitoring

- Permanent monitoring of area F (e.g. temperature profiles, intrusion detection, ...)
- Spreading of N sensor nodes
  - ↪ More sensors than necessary for full coverage of F
- At each point in time, activate only as many sensors as necessary
  - ↪ maximize lifetime T

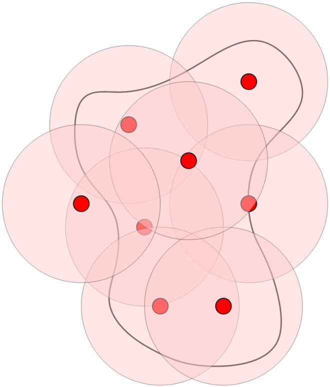


Overview

Problem Formulation - 1

Area Monitoring

- ≫ Permanent monitoring of area F (e.g. temperature profiles, intrusion detection, ...)
- ≫ Spreading of N sensor nodes
  - ↪ More sensors than necessary for full coverage of F
- ≫ At each point in time, activate only as many sensors as necessary
  - ↪ maximize lifetime T

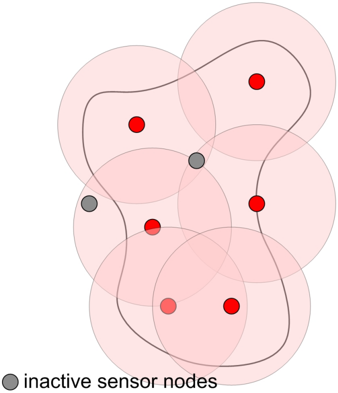


Overview

Problem Formulation - 1

Area Monitoring

- Permanent monitoring of area F (e.g. temperature profiles, intrusion detection, . . .)
- Spreading of N sensor nodes
  - ↪ More sensors than necessary for full coverage of F
- At each point in time, activate only as many sensors as necessary
  - ↪ maximize lifetime T



Overview

Problem Formulation - 2

Problem Denotation

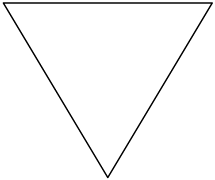
Scheduling of nodes for **L**ifetime maximization of area **C**overage **(SLC)**
(see [BermanCa04] for previous work)

Example

- ≫ Sensors A, B, C with capacity of 1
- ≫ 3 possible covers: AB, BC, AC

(a) Let AB be active for $t = 1$
$\hookrightarrow$ at $T = 1.0$, no further covers possible

(b) Let AB be active for $t = 0.5$,
then, let BC active for $t = 0.5$,
then, let CA active for $t = 0.5$
$\hookrightarrow$ $T = 1.5$, lifetime increased by 50%



Overview

Problem Formulation - 2

Problem Denotation

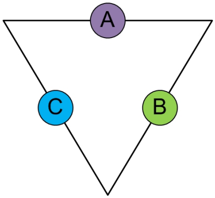
Scheduling of nodes for **L**ifetime maximization of area **C**overage **(SLC)**
(see [BermanCa04] for previous work)

Example

- ≫ Sensors A, B, C with capacity of 1
- ≫ 3 possible covers: AB, BC, AC

(a) Let AB be active for $t = 1$
$\hookrightarrow$ at $T = 1.0$, no further covers possible

(b) Let AB be active for $t = 0.5$,
then, let BC active for $t = 0.5$,
then, let CA active for $t = 0.5$
$\hookrightarrow$ $T = 1.5$, lifetime increased by 50%



Overview

Problem Formulation - 2

Problem Denotation

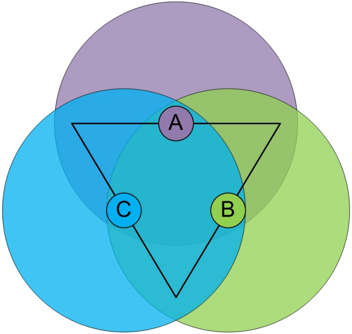
Scheduling of nodes for **L**ifetime maximization of area **C**overage **(SLC)**
(see [BermanCa04] for previous work)

Example

- Sensors A, B, C with capacity of 1
- 3 possible covers: AB, BC, AC

(a) Let AB be active for $t = 1$
$\hookrightarrow$ at $T = 1.0$, no further covers possible

(b) Let AB be active for $t = 0.5$,
then, let BC active for $t = 0.5$,
then, let CA active for $t = 0.5$
$\hookrightarrow$ $T = 1.5$, lifetime increased by 50%



Overview

Problem Formulation - 2

Problem Denotation

Scheduling of nodes for **L**ifetime maximization of area **C**overage **(SLC)**
(see [BermanCa04] for previous work)

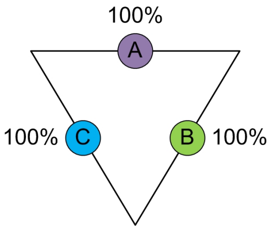
Example

- ≫ Sensors A, B, C with capacity of 1
- ≫ 3 possible covers: AB, BC, AC

(a) Let AB be active for $t = 1$
$\hookrightarrow$ at $T = 1.0$, no further covers possible

(b) Let AB be active for $t = 0.5$,
then, let BC active for $t = 0.5$,
then, let CA active for $t = 0.5$
$\hookrightarrow$ $T = 1.5$, lifetime increased by 50%

(a) trivial solution



Overview

Problem Formulation - 2

Problem Denotation

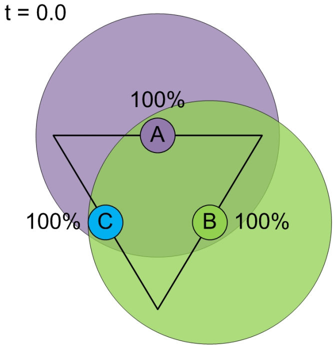
Scheduling of nodes for **L**ifetime maximization of area **C**overage **(SLC)**
(see [BermanCa04] for previous work)

Example

- ≫ Sensors A, B, C with capacity of 1
- ≫ 3 possible covers: AB, BC, AC

(a) Let AB be active for $t = 1$
$\hookrightarrow$ at $T = 1.0$, no further covers possible

(b) Let AB be active for $t = 0.5$,
then, let BC active for $t = 0.5$,
then, let CA active for $t = 0.5$
$\hookrightarrow$ $T = 1.5$, lifetime increased by 50%



Overview

Problem Formulation - 2

Problem Denotation

Scheduling of nodes for **L**ifetime maximization of area **C**overage **(SLC)**
(see [BermanCa04] for previous work)

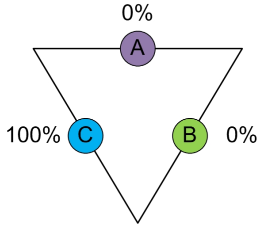
Example

- ≫ Sensors A, B, C with capacity of 1
- ≫ 3 possible covers: AB, BC, AC

(a) Let AB be active for $t = 1$
$\hookrightarrow$ at $T = 1.0$, no further covers possible

(b) Let AB be active for $t = 0.5$,
then, let BC active for $t = 0.5$,
then, let CA active for $t = 0.5$
$\hookrightarrow$ $T = 1.5$, lifetime increased by 50%

t = 1.0



Overview

Problem Formulation - 2

Problem Denotation

Scheduling of nodes for **L**ifetime maximization of area **C**overage **(SLC)**
(see [BermanCa04] for previous work)

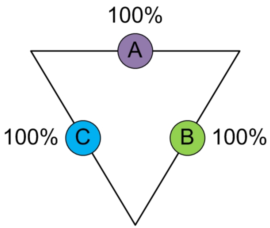
Example

- Sensors A, B, C with capacity of 1
- 3 possible covers: AB, BC, AC

(a) Let AB be active for $t = 1$
$\hookrightarrow$ at $T = 1.0$, no further covers possible

(b) Let AB be active for $t = 0.5$,
then, let BC active for $t = 0.5$,
then, let CA active for $t = 0.5$
$\hookrightarrow$ $T = 1.5$, lifetime increased by 50%

(b) optimal solution



Overview

Problem Formulation - 2

Problem Denotation

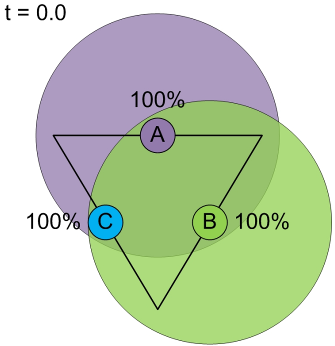
Scheduling of nodes for **L**ifetime maximization of area **C**overage **(SLC)**
(see [BermanCa04] for previous work)

Example

- ≫ Sensors A, B, C with capacity of 1
- ≫ 3 possible covers: AB, BC, AC

(a) Let AB be active for $t = 1$
$\hookrightarrow$ at $T = 1.0$, no further covers possible

(b) Let AB be active for $t = 0.5$,
then, let BC active for $t = 0.5$,
then, let CA active for $t = 0.5$
$\hookrightarrow$ $T = 1.5$, lifetime increased by 50%



Overview

Problem Formulation - 2

Problem Denotation

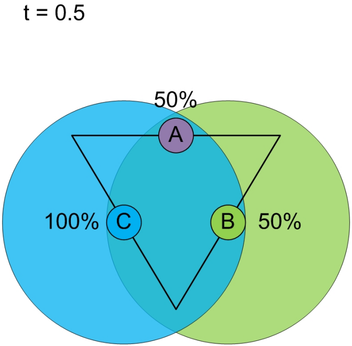
Scheduling of nodes for **L**ifetime maximization of area **C**overage **(SLC)**
(see [BermanCa04] for previous work)

Example

- ≫ Sensors A, B, C with capacity of 1
- ≫ 3 possible covers: AB, BC, AC

(a) Let AB be active for $t = 1$
$\hookrightarrow$ at $T = 1.0$, no further covers possible

(b) Let AB be active for $t = 0.5$,
then, let BC active for $t = 0.5$,
then, let CA active for $t = 0.5$
$\hookrightarrow$ $T = 1.5$, lifetime increased by 50%



Overview

Problem Formulation - 2

Problem Denotation

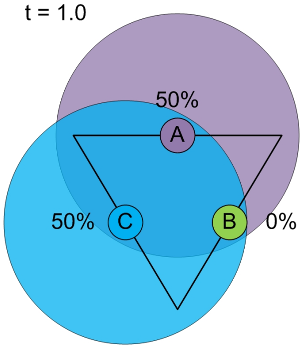
Scheduling of nodes for **L**ifetime maximization of area **C**overage **(SLC)**
(see [BermanCa04] for previous work)

Example

- ≫ Sensors A, B, C with capacity of 1
- ≫ 3 possible covers: AB, BC, AC

(a) Let AB be active for $t = 1$
$\hookrightarrow$ at $T = 1.0$, no further covers possible

(b) Let AB be active for $t = 0.5$,
then, let BC active for $t = 0.5$,
then, let CA active for $t = 0.5$
$\hookrightarrow$ $T = 1.5$, lifetime increased by 50%



Overview

Problem Formulation - 2

Problem Denotation

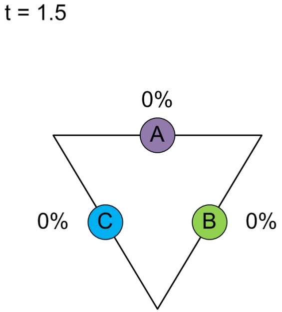
Scheduling of nodes for **L**ifetime maximization of area **C**overage **(SLC)**
(see [BermanCa04] for previous work)

Example

- Sensors A, B, C with capacity of 1
- 3 possible covers: AB, BC, AC

(a) Let AB be active for $t = 1$
$\hookrightarrow$ at $T = 1.0$, no further covers possible

(b) Let AB be active for $t = 0.5$,
then, let BC active for $t = 0.5$,
then, let CA active for $t = 0.5$
$\hookrightarrow$ $T = 1.5$, lifetime increased by 50%

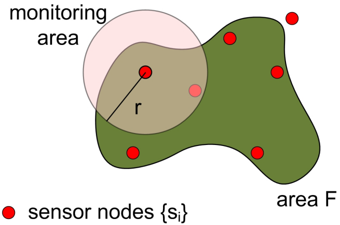


Problem Formulation

Model Description

Given:

- arbitrary area F
- N sensor nodes $S = \{s_i\}$, with
  - fixed position in or near area F
  - circular monitoring area (radius r)
  - limited capacity c_i



Wanted:

- Maximum time T, the whole area can be monitored (lifetime)
- Feasible solutions include:
  - grouping of sensors in M covers $\{\mathcal{C}_j\}$, monitoring the whole area
  - durations $\{t_j\}$, for which each cover $\mathcal{C}_j$ is active (scheduling)

Problem Formulation

Formulation with Linear Programming (LP)

maximize: lifetime

$$T = max\{\mathbf{1}^T\mathbf{t}|\mathbf{t} \in \mathbb{R}^M\}$$

subject to: limited node capacities

$$\sum_{j=1}^{M} A_{i,j}\ t_j \leq c_i \qquad i = 1, \dots N$$

- t_j: duration for which cover $\mathcal{C}_j$ is active
- $A_{i,j}$: 1, if node s_i in cover $\mathcal{C}_j$ is active, 0 otherwise
- c_i: capacity of node s_i

Linear Programming

Useful Attributes

Dual problem

For each primal problem

$$max\{\mathbf{1}^T\mathbf{t} | \mathbf{A}\mathbf{t} \leq \mathbf{c}, \mathbf{t} \in \mathbb{R}^M\}$$

there is a dual problem

$$min\{\mathbf{c}^T\mathbf{w} | \mathbf{A}^T\mathbf{w} \geq \mathbf{1}, \mathbf{w} \in \mathbb{R}^N\}$$

≫ w_i: newly introduced variables by dual problem,
interpretation in context of SLC: "cost" of node s_i

Hardness of the Problem

Sketch of the Proof of NP-Completeness - 1

Utilized Problems

(1) Separation problem for dual problem of SLC (**SEP**):
same complexity as primal problem, see [GrötschelLoSc81]

Given w, does a cover $\mathcal{C}_j$ exist with cost $\sum_{i,\ A_{i,j}=1} w_i < b_j$?

(2) Minimum Dominating Set (**MDS**) on Unit Disk Graphs (**UD**):
proven to be NP-hard, see [MasuyamaIbHa81]

Given a unit disk graph $G = (S, E)$, find $D \subseteq S$ with $|D|$ minimal and f.a. $d \in D$: $d \in S$ or $(d, s) \in E$ with $s \in S$

Hardness of the Problem

Sketch of the Proof of NP-Completeness - 2

Basic Ideas

- MDS-UD can be interpreted as special case of SEP
  - equal costs for all nodes
  - area coverage → point coverage
  - sensor networks → unit disk graphs
  - sensor positions as points to be covered (dominated)

⇒ SLC is NP-hard

- A potential solution can be verified in polynomial time

⇒ SLC is NP-complete

Exact Algorithm

Prelimenaries

Naive Idea

- Use LP formulation with LP solver (e.g. CPLEX)

Problems

- matrix **A** for all possible covers is exponential in size
- actually required covers $\mathcal{C}_j$ not known a priori

Solution

- Column Generation Technique **(CGT)**

Exact Algorithm

Column Generation Technique - 1

Definition

- Let $\mathbf{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_M\}$ be the constraint matrix
  $\hookrightarrow$ each $\mathbf{a}_i$ represents a cover

Iterative Process

- Solve reduced problem with $\hat{\mathbf{A}} \subset \mathbf{A}$
  - fewer possible covers available
  - primal & dual solutions: $\hat{\mathbf{t}}$, $\hat{\mathbf{w}}$
- Solve subproblem: $W = min\{\mathbf{a}^T\hat{\mathbf{w}} - 1 | \mathbf{a} \in \mathbf{A}\}$
  - if $W \geq 0$, $\hat{\mathbf{t}}$ optimal solution for original problem,
  - otherwise, $\mathbf{a}$ provides a new column for $\hat{\mathbf{A}}$

Exact Algorithm

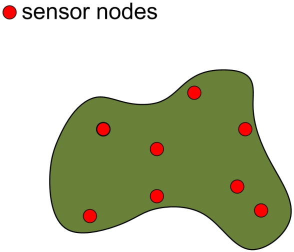
Column Generation Technique - 2

Subproblem

$$W = min\{\mathbf{a}^T \hat{\mathbf{w}} - 1 | \mathbf{a} \in \mathbf{A}\}$$

- **a**: cover of area F
- $\hat{\mathbf{w}}$: weights of each sensor

- equivalent to Min-Cost Set Cover
  - still NP-hard,
  - but many exisiting solvers



Results

- large candidate set of covers becomes manageable with CGT

Exact Algorithm

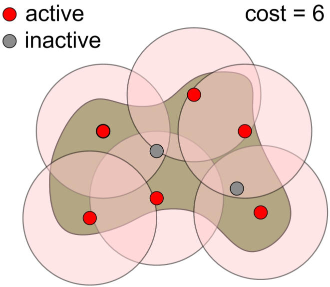
Column Generation Technique - 2

Subproblem

$$W = min\{\mathbf{a}^T\hat{\mathbf{w}} - 1 | \mathbf{a} \in \mathbf{A}\}$$

- $\mathbf{a}$: cover of area F
- $\hat{\mathbf{w}}$: weights of each sensor

- equivalent to Min-Cost Set Cover
  - still NP-hard,
  - but many exisiting solvers



Results

- large candidate set of covers becomes manageable with CGT

Exact Algorithm

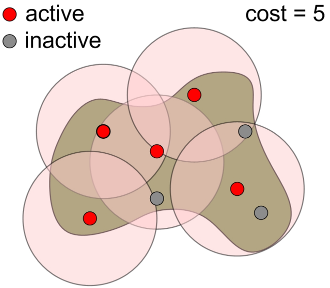
Column Generation Technique - 2

Subproblem

$$W = min\{\mathbf{a}^T \hat{\mathbf{w}} - 1 | \mathbf{a} \in \mathbf{A}\}$$

- **a**: cover of area F
- $\hat{\mathbf{w}}$: weights of each sensor

- equivalent to Min-Cost Set Cover
  - still NP-hard,
  - but many exisiting solvers



Results

- large candidate set of covers becomes manageable with CGT

Approximation Algorithm

Prelimenaries

Some Definitions

- Let T_r be a feasible solution of an SLC instance with sensor radii r,
- let $T_r = opt_r$ be the optimal solution

Approach

- Relax two attributes to provide a fast approximation algorithm for the SLC problem
  - sensor radii r
  - maximum lifetime T

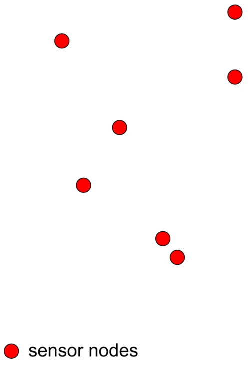
Approximation Algorithm

Approach - First Relaxation

Sensor Radii

- Relocation of all sensor nodes to a grid of size $r \cdot \delta/2$
- Let algorithm $\mathcal{A}$ provide an α-approximation for this problem
  - $\rightarrow$ $\mathcal{A}$ yields solution for the general problem with $T_r \geq \alpha \cdot opt_{(1-\delta)r}$

- Relaxation of sensor radii:
  $\hookrightarrow$ reduction by a factor of $(1-\delta)$



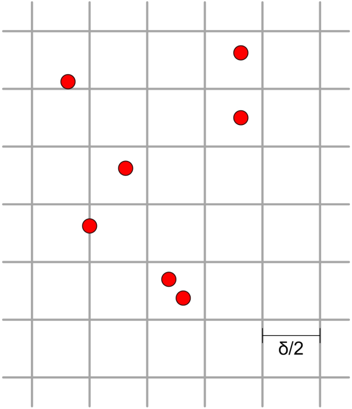
Approximation Algorithm

Approach - First Relaxation

Sensor Radii

- Relocation of all sensor nodes to a grid of size $r \cdot \delta/2$
- Let algorithm $\mathcal{A}$ provide an α-approximation for this problem
  - → $\mathcal{A}$ yields solution for the general problem with $T_r \geq \alpha \cdot opt_{(1-\delta)r}$

- Relaxation of sensor radii:
  ↪ reduction by a factor of $(1-\delta)$



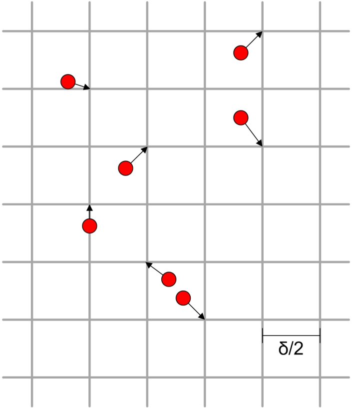
Approximation Algorithm

Approach - First Relaxation

Sensor Radii

- Relocation of all sensor nodes to a grid of size $r \cdot \delta/2$
- Let algorithm $\mathcal{A}$ provide an α-approximation for this problem
  - → $\mathcal{A}$ yields solution for the general problem with $T_r \geq \alpha \cdot opt_{(1-\delta)r}$

- Relaxation of sensor radii:
  ↪ reduction by a factor of $(1 - \delta)$



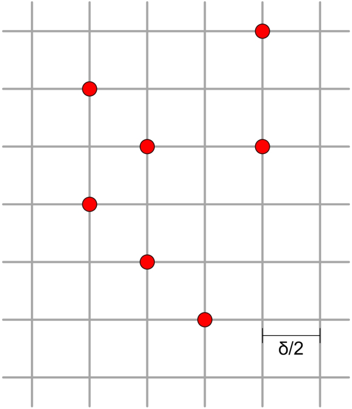
Approximation Algorithm

Approach - First Relaxation

Sensor Radii

- Relocation of all sensor nodes to a grid of size $r \cdot \delta/2$
- Let algorithm $\mathcal{A}$ provide an α-approximation for this problem
  - $\rightarrow$ $\mathcal{A}$ yields solution for the general problem with $T_r \geq \alpha \cdot opt_{(1-\delta)r}$

- Relaxation of sensor radii:
  $\hookrightarrow$ reduction by a factor of $(1-\delta)$



Approximation Algorithm

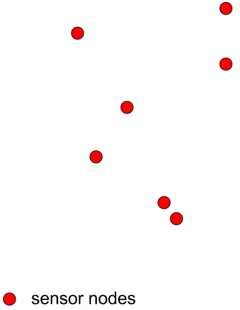
Approach - Second Relaxation - 1

Maximum Lifetime

- Generate tiling $\mathcal{T}$ of area $\mathcal{F}$ in squares of width $k = \lceil 10/\epsilon \rceil$
- Generate shiftings $\mathcal{T}_i$ of $\mathcal{T}$ by (i, i) with $i \in \mathbb{Z}_k$

Observations for $\mathbf{r = 1}$:

- each monitoring area
  - is cut by at most 2 of the tilings $\mathcal{T}_i$,
  - intersects at most 4 squares



Approximation Algorithm

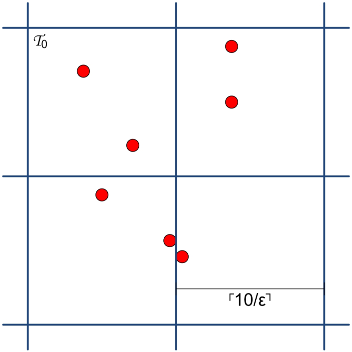
Approach - Second Relaxation - 1

Maximum Lifetime

- Generate tiling $\mathcal{T}$ of area $\mathcal{F}$ in squares of width $k = \lceil 10/\epsilon \rceil$
- Generate shiftings $\mathcal{T}_i$ of $\mathcal{T}$ by (i, i) with $i \in \mathbb{Z}_k$

Observations for r = 1:

- each monitoring area
  - is cut by at most 2 of the tilings $\mathcal{T}_i$,
  - intersects at most 4 squares



Approximation Algorithm

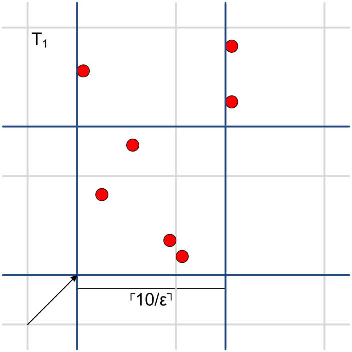
Approach - Second Relaxation - 1

Maximum Lifetime

- Generate tiling $\mathcal{T}$ of area $\mathcal{F}$ in squares of width $k = \lceil 10/\epsilon \rceil$
- Generate shiftings $\mathcal{T}_i$ of $\mathcal{T}$ by (i, i) with $i \in \mathbb{Z}_k$

Observations for r = 1:

- each monitoring area
  - is cut by at most 2 of the tilings $\mathcal{T}_i$,
  - intersects at most 4 squares



Approximation Algorithm

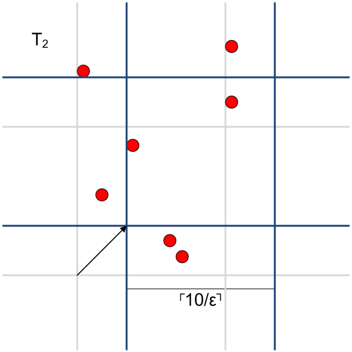
Approach - Second Relaxation - 1

Maximum Lifetime

- Generate tiling $\mathcal{T}$ of area $\mathcal{F}$ in squares of width $k = \lceil 10/\epsilon \rceil$
- Generate shiftings $\mathcal{T}_i$ of $\mathcal{T}$ by (i, i) with $i \in \mathbb{Z}_k$

Observations for r = 1:

- each monitoring area
  - is cut by at most 2 of the tilings $\mathcal{T}_i$,
  - intersects at most 4 squares

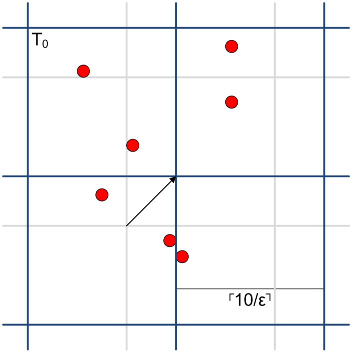


Approximation Algorithm

Approach - Second Relaxation - 1

Maximum Lifetime

- Generate tiling $\mathcal{T}$ of area $\mathcal{F}$ in squares of width $k = \lceil 10/\epsilon \rceil$
- Generate shiftings $\mathcal{T}_i$ of $\mathcal{T}$ by (i, i) with $i \in \mathbb{Z}_k$



Observations for r = 1:

- each monitoring area
  - is cut by at most 2 of the tilings $\mathcal{T}_i$,
  - intersects at most 4 squares

Approximation Algorithm

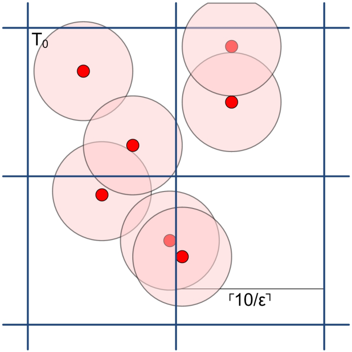
Approach - Second Relaxation - 1

Maximum Lifetime

- Generate tiling $\mathcal{T}$ of area $\mathcal{F}$ in squares of width $k = \lceil 10/\epsilon \rceil$
- Generate shiftings $\mathcal{T}_i$ of $\mathcal{T}$ by (i, i) with $i \in \mathbb{Z}_k$

Observations for r = 1:

- each monitoring area
  - is cut by at most 2 of the tilings $\mathcal{T}_i$,
  - intersects at most 4 squares



Approximation Algorithm

Approach - Second Relaxation - 2

- Let algorithm $\mathcal{A}$ provide an α-approximation for instances of SLC restricted to an area of size $k \times k$
  - Run $\mathcal{A}$ on each square of $\mathcal{T}_i$; yields solution for F with:
    - $T_1 = \alpha \cdot opt_1$
    - at most 4x excess use of each node
  - Combine solutions $\{t_j\}_i$ of all $\mathcal{T}_i$ according to $\{t_j\} = \frac{1-\epsilon}{k} \sum_{i \in \mathbb{Z}_k} \{t_j\}_i$; yields overall solution for F with:
    - $T_1 = (1-\epsilon) \cdot \alpha \cdot opt_1$
    - no violation of capacity constraints

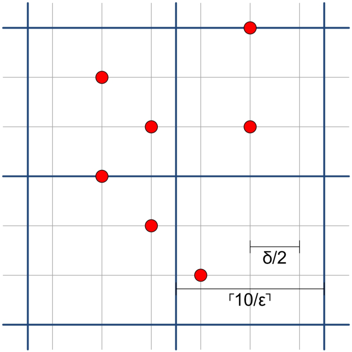
- Relaxation of maximum lifetime:
  $\hookrightarrow$ reduction by a factor of $(1-\epsilon)$

Approximation Algorithm

Approach - Joined Approximations

Combination of both Relaxations

- Let $\mathcal{A}$ be an algorithm that provides an α-approximation of SLC for
  - squared areas of width $k \times k$, and
  - sensor positions restricted to a grid of size $\delta/2$



Observation:

- Each tile has to consider at most $O(1/\delta^2\epsilon^2)$ sensor nodes

→ independent of N !

Approximation Algorithm

Results - 1

Approximation guarantee

$$T_1 \geq (1 - \epsilon) \cdot \alpha \cdot opt_{1-\delta}$$

- $(1 - \epsilon)$: Segmentation of area F into smaller tiles
- α: Approximation guarantee of algorithm $\mathcal{A}$
- $opt_{1-\delta}$: Restriction of sensor positions to a grid

Applied relaxations:

- Actual sensor radii are allowed to be smaller than r
- Maximum lifetime T is allowed to be smaller than the optimum

Approximation Algorithm

Results - 2

Asymptotic running time

$$O\left(N + 1/\epsilon \cdot \frac{\epsilon^2 \cdot N}{opt_{1-\delta}} \cdot f\left(O\left(1/\delta^2\epsilon^2\right)\right)\right)$$

- $O(N)$: Costs for relocation of sensor nodes to grid points
- $O(1/\epsilon)$: Number of tilings $\mathcal{T}_i$ of area F
- $O(\frac{\epsilon^2 \cdot N}{opt_{1-\delta}})$: Number of tiles to be considered per tiling
- $O\left(f\left(O\left(1/\delta^2\epsilon^2\right)\right)\right)$: Running time of algorithm $\mathcal{A}$

Remarks:

- Running time is linear in N
- $\mathcal{A}$ can even take exponential time, since independent of N

– II –
Conclusion
Summary and Outlook

Summary and Outlook
Area Monitoring

Summary
- Proof of NP completeness
- Framework for exact algorithm
- Linear-time approximation scheme

Outlook
- Implementation of both algorithms
- Generalisation to arbitrary (convex) monitoring areas and general metriks (David Steurer - Princeton University)

Time for questions

Thank you,
for your attention!

References

[BermanCa04] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky, *Power efficient monitoring management in sensor networks*, in "Wireless Communications and Networking Conference", 2004

[GrötschelLoSc81] M. Grötschel, L. Lovász, and A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, in "Combinatorica 1 (1981) no. 2", 1981

[MasuyamaIbHa81] S. Masuyama, T. Ibaraki, and T. Hasegawa, *The computational complexity of the m-center problems on the plane*, in "IEICE Transactions 64 (1981) no. 2", 1981