Distributed computing of coverage in sensor networks by homological methods

Paweł Dłotko and Mateusz Juda
(joint work with Robert Ghrist and Marian Mrozek)

Jagiellonian University, Krakow

22 June 2010

Formulation of the problem.

- By a *sensor* we mean a small device designed to sense the presence of some objects.

Formulation of the problem.

- By a *sensor* we mean a small device designed to sense the presence of some objects.
- Sensors do not have any metric information.

Formulation of the problem.

- By a *sensor* we mean a small device designed to sense the presence of some objects.
- Sensors do not have any metric information.
- Each sensor is equipped with a processor and radio transmitter. The radio transmitter gives the sensor the ability to communicate with nearby sensors.

Formulation of the problem.

- By a *sensor network* we mean a triple (S, r_b, r_c) where:

Formulation of the problem.

- By a *sensor network* we mean a triple (S, r_b, r_c) where:
    1. $S \subset \mathbb{R}^2$ is the set of location points of a finite family of identical sensors distributed over the plane,

Formulation of the problem.

- By a *sensor network* we mean a triple (S, r_b, r_c) where:
  1. $S \subset \mathbb{R}^2$ is the set of location points of a finite family of identical sensors distributed over the plane,
  2. r_b is the maximal distance in which each sensor is able to communicate,

Formulation of the problem.

- By a *sensor network* we mean a triple (S, r_b, r_c) where:
  1. $S \subset \mathbb{R}^2$ is the set of location points of a finite family of identical sensors distributed over the plane,
  2. r_b is the maximal distance in which each sensor is able to communicate,
  3. r_c is the maximal distance in which each sensor is able to sense the presence of objects (the radius of coverage).

Formulation of the problem.

- By a *sensor network* we mean a triple (S, r_b, r_c) where:
  1. $S \subset \mathbb{R}^2$ is the set of location points of a finite family of identical sensors distributed over the plane,
  2. r_b is the maximal distance in which each sensor is able to communicate,
  3. r_c is the maximal distance in which each sensor is able to sense the presence of objects (the radius of coverage).
  4. $r_c \geq \frac{r_b}{\sqrt{3}}$

Formulation of the problem.

- By a *sensor network* we mean a triple (S, r_b, r_c) where:
  1. $S \subset \mathbb{R}^2$ is the set of location points of a finite family of identical sensors distributed over the plane,
  2. r_b is the maximal distance in which each sensor is able to communicate,
  3. r_c is the maximal distance in which each sensor is able to sense the presence of objects (the radius of coverage).
  4. $r_c \geq \frac{r_b}{\sqrt{3}}$
- One of the fundamental problems concerning sensor networks is determining whether every point in a given bounded set $\mathcal{D} \subset \mathbb{R}^2$ is covered by the network, i.e. if

$$\mathcal{D} \subset \bigcup_{s \in S} B(s, r_c).$$

- The importance of the coverage problem in the sensor networks: security, mine field sweeping, ad hoc communication networks.

Assumptions.

- A *sensor network graph* associated with a sensor network (S, r_b, r_c) is a graph whose vertices coincide with S and whose edges are doubletons $\{s_1, s_2\} \in \binom{S}{2}$ such that $d(s_1, s_2) \leq r_b$. We denote such a graph by $G_{(S, r_b, r_c)}$.

Assumptions.

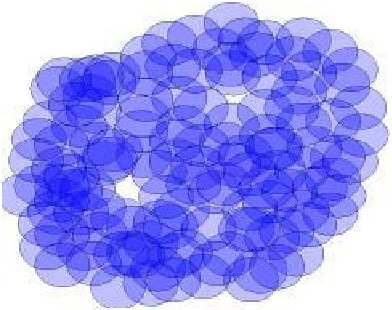
- A *sensor network graph* associated with a sensor network (S, r_b, r_c) is a graph whose vertices coincide with S and whose edges are doubletons $\{s_1, s_2\} \in \binom{S}{2}$ such that $d(s_1, s_2) \leq r_b$. We denote such a graph by $G_{(S,r_b,r_c)}$.
- By a *fence* of a sensor network (S, r_b, r_c) we mean a subset $F \subset S$ such that the geometric realization of the subgraph of $G_{(S,r_b,r_c)}$ induced by F is a Jordan curve in $\mathbb{R}^2$.

Assumptions.

- A *sensor network graph* associated with a sensor network (S, r_b, r_c) is a graph whose vertices coincide with S and whose edges are doubletons $\{s_1, s_2\} \in \binom{S}{2}$ such that $d(s_1, s_2) \leq r_b$. We denote such a graph by $G_{(S,r_b,r_c)}$.
- By a *fence* of a sensor network (S, r_b, r_c) we mean a subset $F \subset S$ such that the geometric realization of the subgraph of $G_{(S,r_b,r_c)}$ induced by F is a Jordan curve in $\mathbb{R}^2$.
- The presented algorithm verifies the partial criterion of the coverage for subsets $\mathcal{D} \subset \mathbb{R}^2$ which constitute the bounded component of the complement of the Jordan curve of a fence.

Example.

Figure: An example of a region covered by a sensor system. [2]



The homological criterion

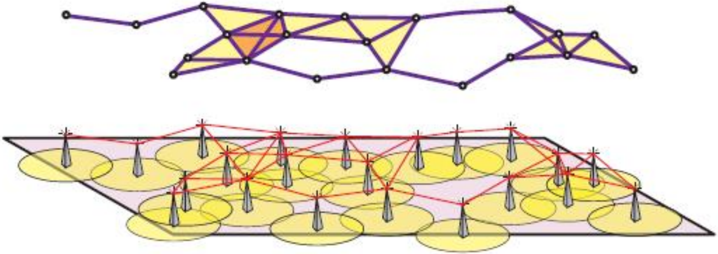
The coverage criterion based on the relative homology of the Rips complexes.

Theorem (Coverage criterion, see [2])

Let (S, r_b, r_c) *be a sensor network with a fence* F *and let* $\mathcal{R} := \mathcal{R}_{r_b}(S)$ *and* $\mathcal{F} := \mathcal{R}_{r_b}(F)$ *be the associated Rips complexes. If there exists a homology class* $[\alpha] \in H_2(\mathcal{R}, \mathcal{F})$ *such that* $\partial\alpha \neq 0$*, then* $\mathcal{D}$*, the bounded component of the Jordan curve of the fence, satisfies*

$$\mathcal{D} \subseteq \bigcup \{B \subset \mathbb{R}^2 \mid B = B(s, r_c), s \in S\}.$$

Example of a Rips complex [4]



What we should do to check this criterion

What we should do to check this criterion

- The sensors need to be distributed in the given region.

What we should do to check this criterion

- The sensors need to be distributed in the given region.
- The fence sensors have to know that they belong to the fence.

What we should do to check this criterion

- The sensors need to be distributed in the given region.
- The fence sensors have to know that they belong to the fence.
- Each sensor has to detect all its neighbors (it suffices for construction of Rips complex).

What we should do to check this criterion

- The sensors need to be distributed in the given region.
- The fence sensors have to know that they belong to the fence.
- Each sensor has to detect all its neighbors (it suffices for construction of Rips complex).
- Construct the Rips complex.

What we should do to check this criterion

- The sensors need to be distributed in the given region.
- The fence sensors have to know that they belong to the fence.
- Each sensor has to detect all its neighbors (it suffices for construction of Rips complex).
- Construct the Rips complex.
- Compute homology of the complex.

What we should do to check this criterion

- The sensors need to be distributed in the given region.
- The fence sensors have to know that they belong to the fence.
- Each sensor has to detect all its neighbors (it suffices for construction of Rips complex).
- Construct the Rips complex.
- Compute homology of the complex.
- The last two things can be done in theory after gathering all the information from all the sensors into one computer...

What we should do to check this criterion

- The sensors need to be distributed in the given region.
- The fence sensors have to know that they belong to the fence.
- Each sensor has to detect all its neighbors (it suffices for construction of Rips complex).
- Construct the Rips complex.
- Compute homology of the complex.
- The last two things can be done in theory after gathering all the information from all the sensors into one computer...
- ...which is not easy and not effective for large complexes.

Idea

- Each sensor hold only “local” information about the Rips complex.

Idea

- Each sensor hold only “local” information about the Rips complex.
- So maybe we can try to make these computations in a distributed way on the sensors themselves?

Idea

- Each sensor hold only “local” information about the Rips complex.
- So maybe we can try to make these computations in a distributed way on the sensors themselves?
- Doing algebraic homology computations in the distributed way is not easy.

Idea

- Each sensor hold only “local” information about the Rips complex.
- So maybe we can try to make these computations in a distributed way on the sensors themselves?
- Doing algebraic homology computations in the distributed way is not easy.
- Why not use the reduction procedures, that can be implemented in a purely combinatorial way?

Idea

- Each sensor hold only “local” information about the Rips complex.
- So maybe we can try to make these computations in a distributed way on the sensors themselves?
- Doing algebraic homology computations in the distributed way is not easy.
- Why not use the reduction procedures, that can be implemented in a purely combinatorial way?
- In some cases the (co)reductions can reduce the complex to homology generators.

How Rips complex can be effectively represented in the sensor network?

- Each sensor $s \in S$ has a unique integer label.

How Rips complex can be effectively represented in the sensor network?

- Each sensor $s \in S$ has a unique integer label.
- If $\sigma = [s_1, ...s_n]$ is a simplex in the Rips complex then the information about σ is kept in the two sensors having the lowest labels among $[s_1, ...s_n]$ (referred to the *lower* and the *upper* sensor controlling σ).

How Rips complex can be effectively represented in the sensor network?

- Each sensor $s \in S$ has a unique integer label.
- If $\sigma = [s_1, ...s_n]$ is a simplex in the Rips complex then the information about σ is kept in the two sensors having the lowest labels among $[s_1, ...s_n]$ (referred to the *lower* and the *upper* sensor controlling σ).
- All decisions about a simplex σ are taken by the lower sensor controlling σ.

Rips complex construction.

Once the sensors are distributed, they need to:

Rips complex construction.

Once the sensors are distributed, they need to:

- detect all their neighbors and the labels of the neighbors of their neighbors,

Rips complex construction.

Once the sensors are distributed, they need to:

- detect all their neighbors and the labels of the neighbors of their neighbors,
- this information is enough to build the Rips complex associated with the sensor network, such that the information about simplex σ is kept in the two sensors,

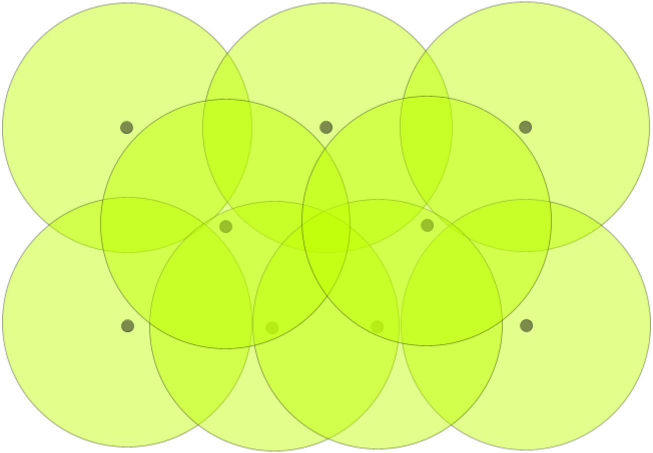
Rips complex construction.

Once the sensors are distributed, they need to:

- detect all their neighbors and the labels of the neighbors of their neighbors,
- this information is enough to build the Rips complex associated with the sensor network, such that the information about simplex σ is kept in the two sensors,
- no further communication is needed to construct the Rips complex.

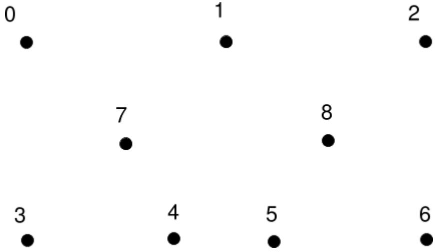
Example.

Figure: An example of a sensor system ($r_c = 2.5$, $r_b = 4$).



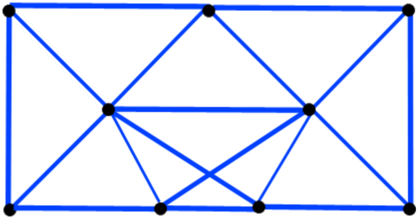
Example.

Figure: 0-simplices.



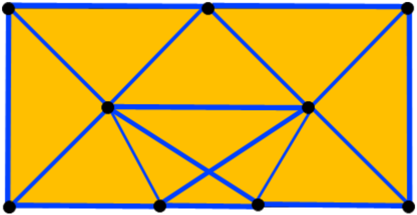
Example.

Figure: 1-simplices.



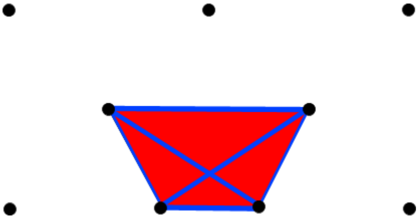
Example.

Figure: 2-simplices.



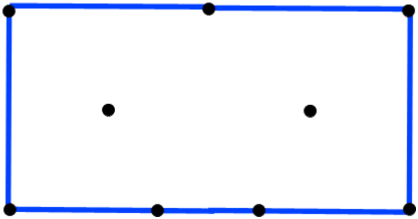
Example.

Figure: 3-simplices.



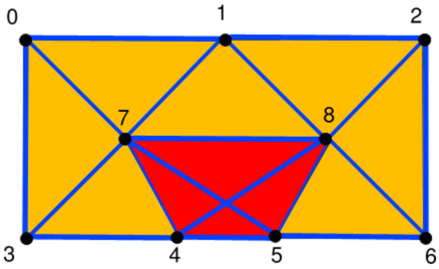
Example.

Figure: Fence subcomplex.



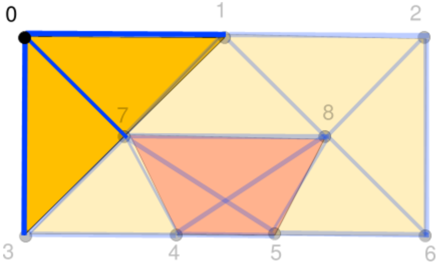
Example.

Figure: All simplices.



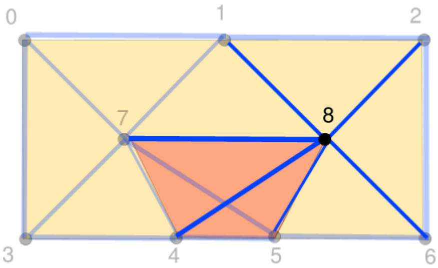
Example.

Figure: Simplices stored in sensor 0.



Example.

Figure: Simplices stored in sensor 8.



Fence reduction procedure

- The coverage criterion needs relative homology group generators $[\alpha] \in H_2(\mathcal{R}, \mathcal{F})$.

Fence reduction procedure

- The coverage criterion needs relative homology group generators $[\alpha] \in H_2(\mathcal{R}, \mathcal{F})$.
- The fence complex $\mathcal{F}$ is a closed subcomplex of the complex $\mathcal{R}$.

Fence reduction procedure

- The coverage criterion needs relative homology group generators $[\alpha] \in H_2(\mathcal{R}, \mathcal{F})$.
- The fence complex $\mathcal{F}$ is a closed subcomplex of the complex $\mathcal{R}$.
- In this case $[\alpha] \in H_2(\mathcal{R}, \mathcal{F})$ is isomorphic to $H_2(\mathcal{R} \backslash \mathcal{F})$.

Fence reduction procedure

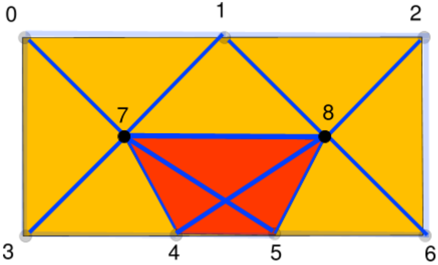
- The coverage criterion needs relative homology group generators $[\alpha] \in H_2(\mathcal{R}, \mathcal{F})$.
- The fence complex $\mathcal{F}$ is a closed subcomplex of the complex $\mathcal{R}$.
- In this case $[\alpha] \in H_2(\mathcal{R}, \mathcal{F})$ is isomorphic to $H_2(\mathcal{R} \backslash \mathcal{F})$.
- So we simply remove the 0 and 1 dimensional simplices that belong to the fence from the constructed Rips complex.

Fence reduction procedure

- The coverage criterion needs relative homology group generators $[\alpha] \in H_2(\mathcal{R}, \mathcal{F})$.
- The fence complex $\mathcal{F}$ is a closed subcomplex of the complex $\mathcal{R}$.
- In this case $[\alpha] \in H_2(\mathcal{R}, \mathcal{F})$ is isomorphic to $H_2(\mathcal{R} \backslash \mathcal{F})$.
- So we simply remove the 0 and 1 dimensional simplices that belong to the fence from the constructed Rips complex.
- This can be done by the sensors themselves.

Example.

Figure: The complex without the fence.



(Co)reduction algorithms

- The created Rips complex is usually a really big structure.

(Co)reduction algorithms

- The created Rips complex is usually a really big structure.
- simplices of arbitrary dimension can appear in it.

(Co)reduction algorithms

- The created Rips complex is usually a really big structure.
- simplices of arbitrary dimension can appear in it.
- This is too large input for the effective homology computation algorithm. We need to reduce the input as much as possible.

(Co)reduction algorithms

- The created Rips complex is usually a really big structure.
- simplices of arbitrary dimension can appear in it.
- This is too large input for the effective homology computation algorithm. We need to reduce the input as much as possible.
- To do so the combination of the reduction and coreduction algorithm will be used.

(Co)reduction algorithms

- A pair (τ, σ) of simplices is called a *reduction pair* if $\text{cbd}\{\tau\} = \{\sigma\}$. A pair (τ, σ) of simplices is called a *coreduction pair* if $\text{bd}\{\sigma\} = \{\tau\}$.

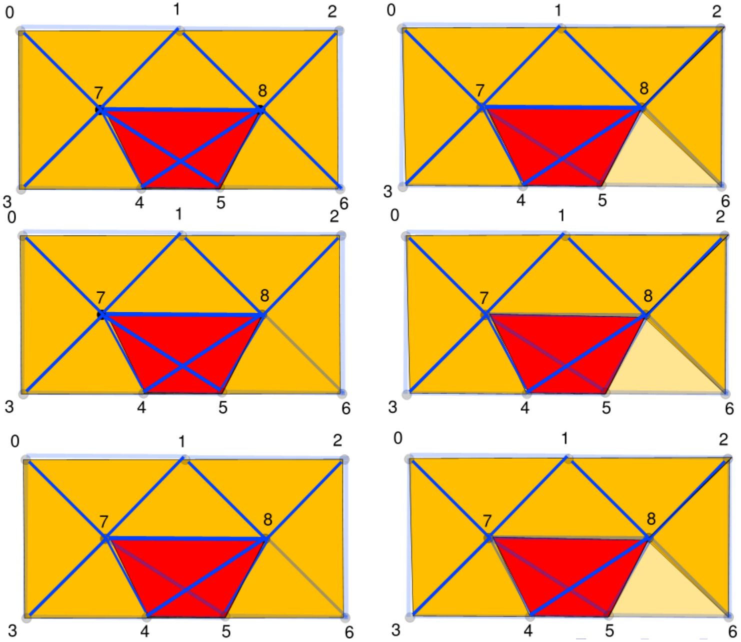
(Co)reduction algorithms

- A pair (τ, σ) of simplices is called a *reduction pair* if $\mathrm{cbd}\{\tau\} = \{\sigma\}$. A pair (τ, σ) of simplices is called a *coreduction pair* if $\mathrm{bd}\{\sigma\} = \{\tau\}$.
- It is shown in [5] that reduction or coreduction pairs may be removed from a complex without changing the homology groups of the complex. We present now how the reductions may be performed in a distributed manner.

(Co)reduction algorithms

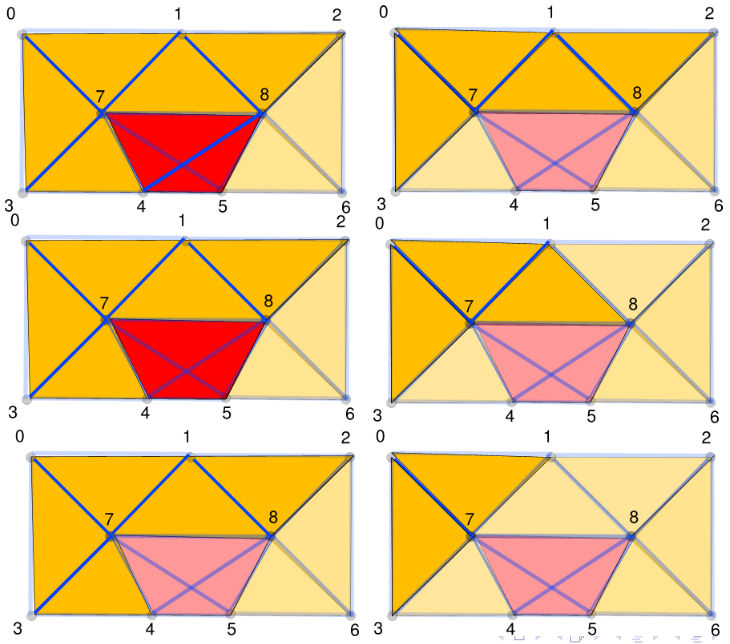
- A pair (τ, σ) of simplices is called a *reduction pair* if $\text{cbd}\{\tau\} = \{\sigma\}$. A pair (τ, σ) of simplices is called a *coreduction pair* if $\text{bd}\{\sigma\} = \{\tau\}$.
- It is shown in [5] that reduction or coreduction pairs may be removed from a complex without changing the homology groups of the complex. We present now how the reductions may be performed in a distributed manner.
- We perform the reduction process in such a way that simplex σ may be reduced only by the lower sensor controlling it.

Example.



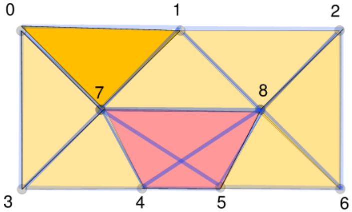
0 1 2
7 8
3 4 5 6
0 1 2
7 8
3 4 5 6
0 1 2
7 8
3 4 5 6
0 1 2
7 8
3 4 5 6
0 1 2
7 8
3 4 5 6
0 1 2
7 8
3 4 5 6

Example.



0
1
2
7
8
3
4
5
6
0
1
2
7
8
3
4
5
6
0
1
2
7
8
3
4
5
6
0
1
2
7
8
3
4
5
6
0
1
2
7
8
3
4
5
6
0
1
2
7
8
3
4
5
6

Example.



Implementation

The algorithms presented in the talk have been implemented in a Sensor Network Simulator. The Sensor Network Simulator executes the presented algorithms on each sensor in a separate thread, so that the simulations may be performed with only a few or even one processor unit.

Efficiency

- Let C_s denote the *CPU time* used by sensor s during the simulation.

Efficiency

- Let C_s denote the *CPU time* used by sensor s during the simulation.
- Let R_s be the *real time* used by sensor s, i.e. the world time passing from the moment the sensor starts the computations until the moment it completes.

Efficiency

- Let C_s denote the *CPU time* used by sensor s during the simulation.
- Let R_s be the *real time* used by sensor s, i.e. the world time passing from the moment the sensor starts the computations until the moment it completes.
- We define the *network time* as $\max_{s \in S} R_s$ and the *running time* as $\sum_{s \in S} C_s$.

Efficiency

- Let C_s denote the *CPU time* used by sensor s during the simulation.
- Let R_s be the *real time* used by sensor s, i.e. the world time passing from the moment the sensor starts the computations until the moment it completes.
- We define the *network time* as $\max_{s \in S} R_s$ and the *running time* as $\sum_{s \in S} C_s$.
- In other words, the network time is the time required by the real network to do the computation. In case of the simulations performed on a single machine with a single CPU, the running time is the CPU usage for the whole simulation.

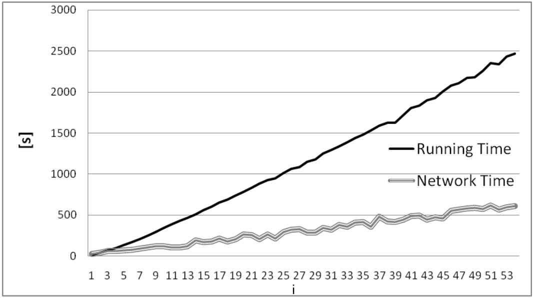
Efficiency

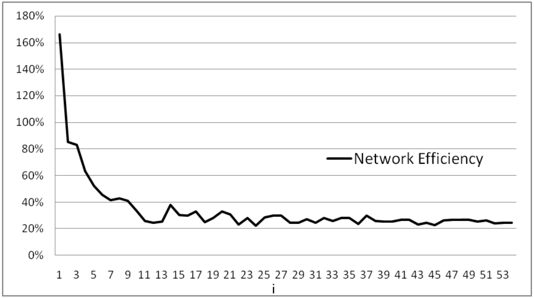
- Let C_s denote the *CPU time* used by sensor s during the simulation.
- Let R_s be the *real time* used by sensor s, i.e. the world time passing from the moment the sensor starts the computations until the moment it completes.
- We define the *network time* as $\max_{s \in S} R_s$ and the *running time* as $\sum_{s \in S} C_s$.
- In other words, the network time is the time required by the real network to do the computation. In case of the simulations performed on a single machine with a single CPU, the running time is the CPU usage for the whole simulation.
- We define the network efficiency as the ratio

$$\texttt{Network Efficiency} := (\texttt{Network Time})/(\texttt{Running Time})\cdot 100\%.$$

Linear test

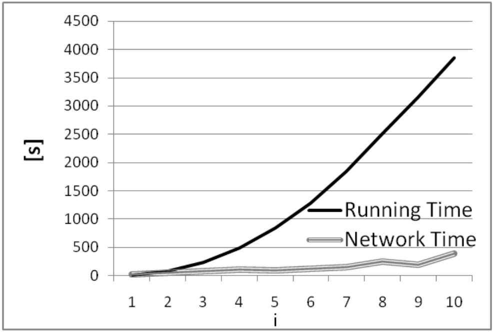
1. L_0 is a base test case consist of 23 sensors randomly placed on a 4×4 units rectangle with the communication radius fixed to 2 units with a rectangular fence around the network,
2. L_i is created from L_{i-1} by placing a new copy of the base test on the right side of L_{i-1} with a new rectangular fence around the network.

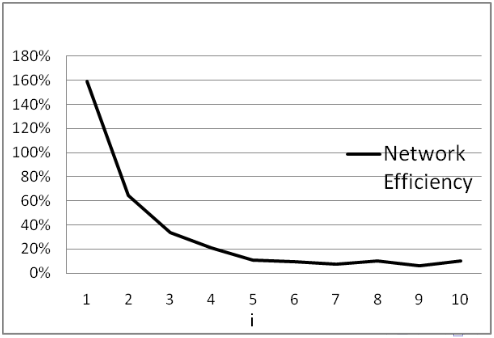

3000
2500
2000
1500
1000
500
0
[s]
Running Time
Network Time
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53
i


180%
160%
140%
120%
100%
80%
60%
40%
20%
0%
Network Efficiency
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53
i

Square test

1. S_0 is the base test case with a rectangular fence around the network,
2. S_i is created from S_{i-1} by placing three new copies of the base test: on the right, on the bottom side, and on the bottom-right side of S_{i-1}. A new rectangular fence around the network is created.


4500
4000
3500
3000
2500
2000
1500
1000
500
0
[s]
Running Time
Network Time
1 2 3 4 5 6 7 8 9 10
i


180%
160%
140%
120%
100%
80%
60%
40%
20%
0%
Network Efficiency
1 2 3 4 5 6 7 8 9 10
i

Can this work? Can we prove that the algorithm is correct?

- In the papers dealing with reduction and coreductions proofs of the correctness of the sequential algorithms provided.
- distributed version of these algorithms used here.
- How to prove the correctness of the distributed version of the algorithm?
- This is what I am going to talk about.

S-complexes, algorithmic reformulation of the chain complexes.

- $\mathcal{R}$ - finite set with gradation,
- $\kappa : \mathcal{R} \times \mathcal{R} \to R$ – incidence map,
- $(\mathcal{R}, \kappa)$ is a S-complex iff $(R(\mathcal{R}), \partial\kappa)$ with $\partial\kappa(\sigma) = \sum_{\tau \in \mathcal{R}} \kappa(\sigma, \tau)\tau$ is a free chain complex with a base $\mathcal{R}$,
- operation done on a basis $\mathcal{R}$, subroutine κ for indices provided,
- a subset $\mathcal{R}' \subset \mathcal{R}$ referred to as regular, if $(\mathcal{R}', \kappa|_{\mathcal{R}' \times \mathcal{R}'})$ is a subcomplex of $(\mathcal{R}, \kappa)$ (well, almost, let's keep it simple to get the idea...),
- after removing a reduction pair from the complex a regular subset is obtained.

Chain maps related to each reduction.

- (τ, σ) – a reduction pair in complex $\mathcal{R}$.
- Map $\eta_{(\tau,\sigma)} : C(\mathcal{R} \setminus \{\tau, \sigma\}) \to C(\mathcal{R})$ inducing isomorphism in homology is given.
- For $c \in C(\mathcal{R} \setminus \{\tau, \sigma\})$, $\eta_{(\tau,\sigma)}(c) = c - \frac{\langle \partial c, \tau \rangle}{\langle \partial \sigma, \tau \rangle} \sigma$.
- Building blocks for the map used in the further formal analysis.

State graph of the S-complex – the formal model.

- $G_{\mathcal{R}} = (V, E)$,
- V consist of regular subsets of $\mathcal{R}$,
- for $\mathcal{K}, \mathcal{K}' \in V$ $(\mathcal{K}, \mathcal{K}') \in E$ iff $\mathcal{K}' = \mathcal{K} \setminus \{\tau, \sigma\}$ for (τ, σ) being a reduction pair in $\mathcal{K}$,
- *Reduction path* ψ - directed path in $G_{\mathcal{R}}$ joining $\mathcal{K}$ and $\mathcal{K}'$,
- need some more specific type of reduction paths to show, that reductions can be applied in a distributed way.

Simple reduction paths.

- Let $\mathcal{A}$ be the set of all reduction pairs in the initial complex $\mathcal{R}$,
- before a reduction of a pair $(\sigma, \tau) \notin \mathcal{A}$ can be made, some of the reductions in $\mathcal{A}$ have to be already made,
- this leads to the idea of simple reduction paths:
- *Reduction path* $(\tau_1, \sigma_1), \ldots, (\tau_n, \sigma_n)$ joining $\mathcal{K}$ and $\mathcal{K}'$ – *simple* if (τ_i, σ_i) is a reduction pair in $\mathcal{K}$ for every $i \in \{1 \ldots n\}$.
- ϕ and ψ s.r.p. sharing the initial vertex - equivalent iff. reductions along ϕ are permutation of reductions along ψ.

Equivalent simple reduction paths induce the same maps in homology.

- $\psi = (\tau_1, \sigma_1) \dots (\tau_n, \sigma_n)$ - simple reduction path between $\mathcal{K}$ and $\mathcal{K}'$,
- induces map in homology $H(\psi) = \eta_{(\tau_1,\sigma_1)} \circ \dots \circ \eta_{(\tau_n,\sigma_n)}$,

- Theorem
  ψ and ϕ – equivalent simple reduction paths joining $\mathcal{K}$ and $\mathcal{K}'$ in the state graph, then $H(\psi) = H(\phi)$,
  - therefore the map in homology does not depend on the choice of a simple reduction path between $\mathcal{K}$ and $\mathcal{K}'$,
  - reductions on the simple reduction paths have disjoint supports,
  - they can be reduced in a distributed way.

Equivalent simple reduction paths induce identical maps in homology.

- Idea of a proof:
- ψ and ϕ differs by a permutation of reduction pairs.
- transpose one of them to get the other one.
- Only the reduction and coreduction pairs considered,
- sufficient to prove for any simple reduction path of length 2 (simple induction).

Distributed reduction model, idea.

- $\mathcal{R}$ – initial, $\mathcal{R}'$ – final complex in the course of reductions,
- usually the reduction process is long and deep,
- consequently no simple reduction path from $\mathcal{R}$ to $\mathcal{R}'$,
- **Idea:** divide, for the sake of proof, the whole execution of the algorithm to some levels,
- $\mathcal{R} \xrightarrow{p_1} \mathcal{K}_1 \xrightarrow{p_2} \cdots \xrightarrow{p_n} \mathcal{R}'$,
- each arrow p_i represents a simple reduction path. In homology induce isomorphism,
- explicit formula for the map p_i and its inverse are given.

Distributed reduction model, aim.

- Suppose, that each sensor stores information for each reduction pair if it was reduced as a reduction or coreduction pair,
- $\mathcal{A}$:=set of all reductions made by all the sensors in the course of running the algorithm,
- aim: show that $H(\mathcal{R} \setminus \mathcal{A}) \backsimeq H(\mathcal{R})$.

Correctness of the reductions via parallel algorithm.

- For $(\tau, \sigma) \in \mathcal{A}$ reduced as coreduction pair

$$\alpha(\tau, \sigma) := \{(\overline{\tau}, \overline{\sigma}) \in \mathcal{A} | \{\overline{\tau}, \overline{\sigma}\} \cap (\mathsf{bd}_{\mathcal{R}}\, \sigma \setminus \tau) \neq \emptyset\},$$

- for $(\tau, \sigma) \in \mathcal{A}$ reduced as reduction pair

$$\alpha(\tau, \sigma) := \{(\overline{\tau}, \overline{\sigma}) \in \mathcal{A} | \{\overline{\tau}, \overline{\sigma}\} \cap (\mathsf{cbd}_{\mathcal{R}}\, \tau \setminus \sigma) \neq \emptyset\},$$

- in "α-sets" reductions that need to be applied prior to (τ, σ) are stored,
- define $\lambda(\tau, \sigma) := 1 + max\{\lambda(\overline{\tau}, \overline{\sigma}) | (\overline{\tau}, \overline{\sigma}) \in \alpha(\tau, \sigma)\}$,
- the function λ is a tool to define the following simple reduction paths.

Correctness of the reductions via parallel algorithm.

- $\mathcal{A}_n := \{(\tau, \sigma) \in \mathcal{A} | \lambda(\tau, \sigma) = n\}$,
- $\mathcal{K}^0 := \mathcal{R}$,
- $\mathcal{K}^n := \mathcal{K}^{n-1} \setminus |\mathcal{A}_n|$,
- **Theorem:** $\mathcal{K}^n$ is an S-complex.
- **Theorem:** if $(\sigma, \tau) \in \mathcal{A}_n$, then (σ, τ) is an reduction or coreduction pair in $\mathcal{K}^{n-1}$,
- global isomorphism between $H(\mathcal{K})$ and $H(\mathcal{K} \setminus \mathcal{A})$ can now be defined as a composition of isomorphisms between $\mathcal{K}^i$ and $\mathcal{K}^{i+1}$ (simple reduction paths!)

The conjecture.

Conjecture

For the relative homology of a Rips complex, built from a planar point set, the sequence of reductions and coreductions reduce the complex up to its homology generators.

- The idea behind this – even if the dimensions of simplices in the Rips complex are high, the topology of it almost "planar" (in the planar case the reductions can do all the job) ...
- this is what we see from numerical experiments,
- even if the Conjecture is not true, the complex that remains after (co)reductions is very small...
- and can be gathered by one of the sensors and the homology can be computed.

Verifying the covering.

- Ghrist–de Silva criterion requires information about the representatives of homology generators in the complex $\mathcal{R}$ (the initial one),
- According to the conjecture we have the homology generator $c \in C_2(\mathcal{R} \setminus \mathcal{A})$,
- information about the applied coreductions allows us to find the isomorphic companion of c in $C_2(\mathcal{R})$,
- explicit formulas for the isomorphisms on the chain level are known,
- they can be implemented as a distributed algorithm,
- this algorithm restores the representant of the generator from $C_2(\mathcal{R} \setminus \mathcal{A})$ to $C_2(\mathcal{R})$.

The bibliography for this section.

- P. D., R. Ghrist, M. J., M. Mrozek, Distributed computing of coverage in sensor networks by homological methods,(in preparation).
- R. Ghrist, A. Muhammad, Coverage and hole-detection in sensor networks via homology, Alg. & Geom. & Topology, (2007). .
- V. de Silva, R. Ghrist, A. Muhammad, Blind Swarms for Coverage in 2-D, Proc. Robotics Systems & Science (2005).
- V. de Silva and R. Ghrist, Coordinate-free coverage in sensor networks with controlled boundaries via homology, Intl. J. Robotics Research (2006).
- B. Batko, M. Mrozek, Coreduction Homology Algorithm, Discrete & Computational Geometry (2009).
- Zin Arai, Kazunori Hayashi, Yasuaki Hiraoka, Mayer-Vietoris sequences and coverage problems in sensor networks, preprint.
- CAPD, $http://capd.ii.uj.edu.pl$.

The end.

Thank you for your attention!