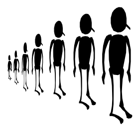
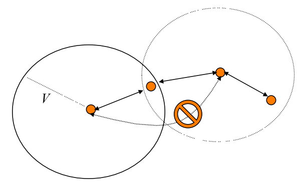
Distributed Coordination of a Set of Autonomous Mobile Robots: The Unlimited Visibility Setting

Giuseppe Prencipe
Dipartimento di Informatica
Università di Pisa
Italy

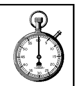
MAC – 15th August 2010



Radius of Visibility: Limited / Unlimited



Timing Models



ASYNC/CORDA - *Fully asynchronous* [Flocchini et. Al, 1999]

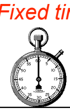
Arbitrary & varying operation rates and delays

SSYNC/SYM - *Semi-synchronous* [Suzuki +Yamashita, 1996]

Fixed time cycles, but robots may be active / inactive

FSYNC/SYM - *Fully synchronous* [Suzuki +Yamashita, 1996]

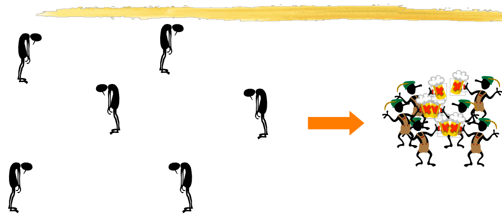
Fixed time cycles, all robots active in every cycle



Problem 1: Gathering,

(Aggregation, rendez-vous, homing)

Gathering - Unlimited Visibility



Initially the robots are in arbitrary distinct positions.

In finite time, they **gather** in the same place.

Motivations

*A conference is a **gathering** of important people who singly can do nothing, but together can decide that nothing can be done*

Fred Allen (1894-1956)

Gathering

- In spite of its apparent simplicity, this problem has been tackled in several studies

Gathering
Unlimited Visibility - SSYNC

Ando, Oasa, Suzuki, Yamashita
Siam Journal Of Computing, 1999

- **Istantaneous activities**
- **n=2, the problem is unsolvable**

Gathering, n=2
Unlimited Visibility - SSYNC

In fact, since the robots have no dimension…

…and cannot bump...

…moving them towards each other is not useful…

…but it works if they can *bump!*

Gathering
Unlimited Visibility - SSYNC

Ando, Oasa, Suzuki, Yamashita
Siam Journal Of Computing, 1999

- **Istantaneous activities**
- **n=2, the problem is unsolvable**
- **n>2, they provide an oblivious algorithm that let the robots gather in finite time**

Gathering
Unlimited Visibility - SSYNC

- MAIN IDEA: Starting from distinct initial positions, we move the robots in such a way that eventually there will be exactly one position that two or more robots occupy
- Each time a robots becomes active, it recognizes the configuration of the robots (7 possible configurations)
- It moves accordingly

Gathering
Unlimited Visibility - SSYNC

Few examples:

- Let C be the smallest enclosing circle of the observed configuration
- If n > 3, and there is exactly one robot r inside C
  - r moves towards one of the robots on the rim of C; the others do not move
- If n > 3, and there are more than two robots inside C
  - The robots inside C move towards the center of C, while the other do not move (the center of C does not change)

Gathering
Unlimited Visibility - SSYNC

- <u>If n > 3, and all robots are on the rim of C</u>
  - All robots move towards the center of C

- In the last case, at the next time instant
  - Either some robots are inside C and some on the rim of C
    - One of the previous cases applies
  - Or all robots are again on the rim of C
    - This same case applies again

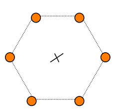
Gathering, ASYNC

- In ASYNC, several factors render this problem difficult to solve

  - Major problems arise from symmetric configurations....

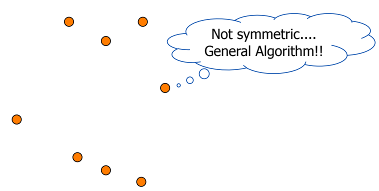
Difficulties

If at the beginning....

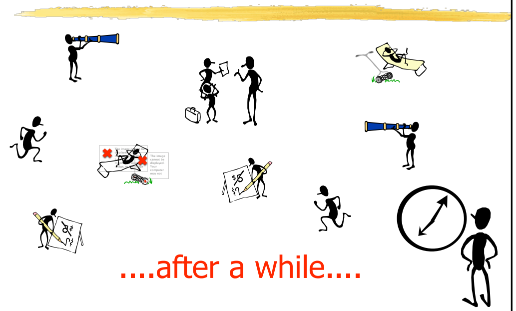


Difficulties

If at the beginning....

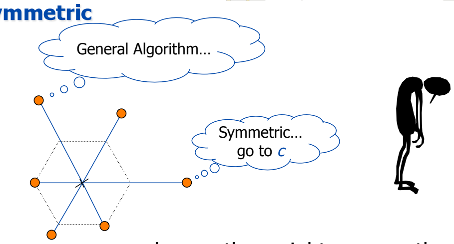


Difficulties



Difficulties

Symmetric



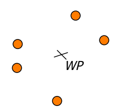
....hence, they might never gather!!!!

Gathering—easy solution

Easy Solution: Weber Point (Weiszfeld, '36)!

Given $r_1,\ldots,r_n$: $WP = \arg\min_{p \in \Re^2} \sum_i \mathrm{dist}(p, r_i)$

1. It is unique (Weiszfeld, '36)

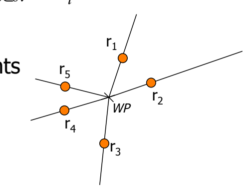


Gathering—easy solution

Easy Solution: Weber Point (Weiszfeld, '36)!

Given $r_1,\ldots,r_n$: $WP = \arg\min_{p \in \Re^2} \sum_i \mathrm{dist}(p, r_i)$

1. It is unique (Weiszfeld, '36)
2. WP is Weber Point of points on $[r_i, WP]$ (Weiszfeld, '36)

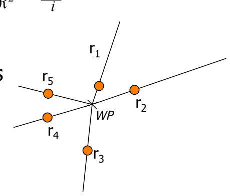


Gathering—easy solution

Easy Solution: Weber Point (Weiszfeld, '36)!

Given $r_1,\ldots,r_n$: $WP = \arg\min_{p \in \Re^2} \sum_i \mathrm{dist}(p, r_i)$

1. It is unique (Weiszfeld, '36)
2. WP is Weber Point of points on $[r_i, WP]$ (Weiszfeld, '36)

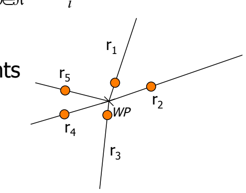


Gathering—easy solution

Easy Solution: Weber Point (Weiszfeld, '36)!

Given $r_1,\ldots,r_n$: $WP = \arg\min_{p \in \Re^2} \sum_i \mathrm{dist}(p, r_i)$

1. It is unique (Weiszfeld, '36)
2. WP is Weber Point of points on $[r_i, WP]$ (Weiszfeld, '36)

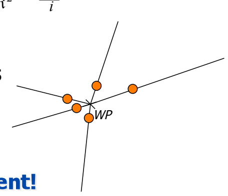


Gathering—easy solution

Easy Solution: Weber Point (Weiszfeld, '36)!

Given $r_1,\ldots,r_n$: $WP = \arg\min_{p \in \Re^2} \sum_i \mathrm{dist}(p, r_i)$

1. It is unique (Weiszfeld, '36)
2. WP is Weber Point of points on $[r_i, WP]$ (Weiszfeld, '36)



WP Invariant Under Movement!

Gathering—easy solution

Easy Solution: Weber Point (Weiszfeld, '36)!

Algorithm:

1. Compute WP
2. Move Towards WP

Unfortunately, WP is not computable!

August 15, 2010 MAC 2010, Ottawa

August 15, 2010 MAC 2010, Ottawa

Gathering
(Unlimited Visibility, no agreement)

n=2: Unsolvable (by Suzuki *et al.*),
unless they can *bump into each other*!

August 15, 2010 MAC 2010, Ottawa

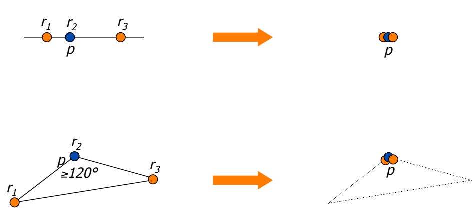
Gathering
(Unlimited Visibility, no agreement)

n=2: Unsolvable (by Suzuki *et al.*),
unless they can *bump into each other*!

n=3,4: Always solvable!

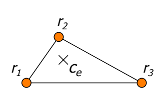
August 15, 2010 MAC 2010, Ottawa

Gathering, *n=3*
(Unlimited Visibility, no agreement)



August 15, 2010 MAC 2010, Ottawa

Gathering, *n=3*
(Unlimited Visibility, no agreement)

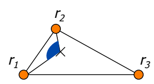


Properties of c_e:

c_e=center of equiangularity

August 15, 2010 MAC 2010, Ottawa

Gathering, *n=3*
(Unlimited Visibility, no agreement)



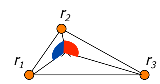
c_e=center of equiangularity

Properties of c_e:

- $r_1\widehat{c_e}r_2$

August 15, 2010 MAC 2010, Ottawa

Gathering, *n=3*
(Unlimited Visibility, no agreement)



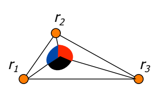
c_e=center of equiangularity

Properties of c_e:

- $r_1\widehat{c_e}r_2 = r_2\widehat{c_e}r_3$

August 15, 2010 MAC 2010, Ottawa

Gathering, *n=3*
(Unlimited Visibility, no agreement)



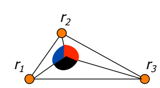
c_e=center of equiangularity

Properties of c_e:

- $r_1\widehat{c_e}r_2 = r_2\widehat{c_e}r_3 = r_3\widehat{c_e}r_1$

August 15, 2010 MAC 2010, Ottawa

Gathering, *n=3*
(Unlimited Visibility, no agreement)



c_e=center of equiangularity

Properties of c_e:

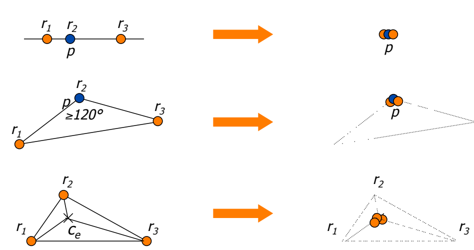
- $r_1\widehat{c_e}r_2 = r_2\widehat{c_e}r_3 = r_3\widehat{c_e}r_1$
- Invariant under movement

August 15, 2010 MAC 2010, Ottawa

Gathering, *n=3*
(Unlimited Visibility, no agreement)

August 15, 2010 MAC 2010, Ottawa

Gathering, *n=3*
(Unlimited Visibility, no agreement)

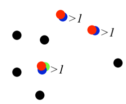


August 15, 2010 MAC 2010, Ottawa

General Schema

- Use of Multiplicity Detection
  - n=3,4 (and even with the use of Weber Point)

Is there 1 or more than 1 robot at a point?



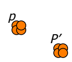
General Schema

The general idea of the solutions is based on multiplicity detection, as follows

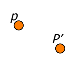
1. At the beginning, robots on distinct positions
2. Get a scenario where there is only one point *p* with multiplicity greater than one
3. All robots move towards *p*

Multiplicity Detection

If the robots cannot detect multiplicities…



is as….

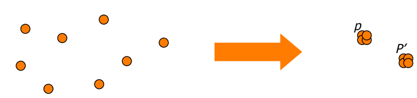


…the proposed solutions do not work!!!!

Multiplicity Detection

For n=2, the problem is not solvable (*Suzuki et al., 1999*)!

It is possible to design an adversary that lets the robots occupy two distinct positions on the plane in a finite number of cycles…



Multiplicity Detection

….hence….

Problem not solvable with n=2

+

No multiplicity detection

⬇

Problem not solvable for any n!

Gathering

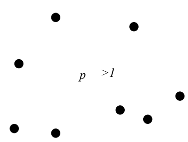
No agreement on the local coordinate systems, and ***oblivious*** robots….

⬇

[TCS 2007]
Necessary Condition: Multiplicity Detection!
(in SSYNC, hence in ASYNC)

The Main Idea (ICALP 2003)

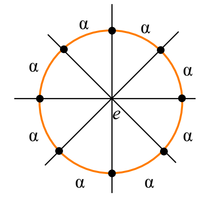
- All distinct positions in initial configuration
- Some robots gather at a point *p*
- Point *p* is unique point with multiplicity >1
- All robots move to *p*



August 15, 2010 MAC 2010, Ottawa

A Special Case

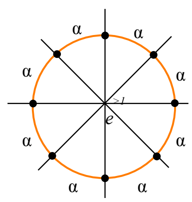
The robots can be in a totally symmetric configuration.



August 15, 2010 MAC 2010, Ottawa

Equiangular Configurations

- All adjacent robots have angle α w.r.t. center e
- Center e is easy to compute
- Move all robots towards e



August 15, 2010 MAC 2010, Ottawa

Algorithm Skeleton

If strict multiplicity: move there

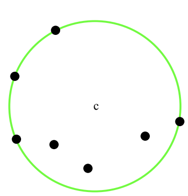
Else If equiangular: move to center

Else elect some robots to gather

August 15, 2010 MAC 2010, Ottawa

Smallest Enclosing Circle

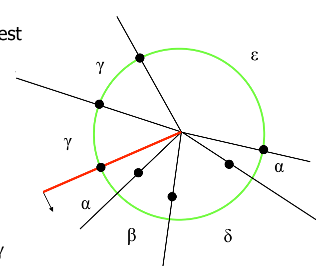
- Unique
- Easy to compute
- Invariant if suitable robots move



August 15, 2010 MAC 2010, Ottawa

String of Angles

- W.r.t. center of smallest enclosing circle
- Circular string
- Can contain 0's

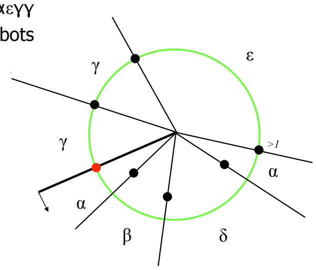


Lex. minimum: αβδαεγγ

August 15, 2010 MAC 2010, Ottawa

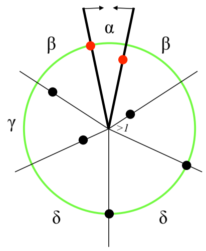
1 Starting Position

- Lex. min string: αβδαεγγ
- Unique ordering of robots



2 Starting Positions

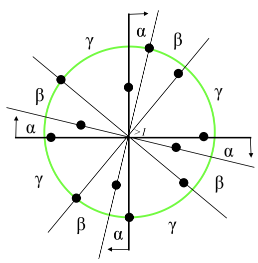
- Palindrome: αβγδδγβ



- or Periodic: αβγδ αβγδ

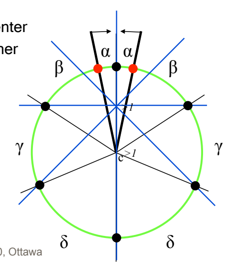
3..*n*-1 Starting Positions

- Periodic: αβγ αβγ …



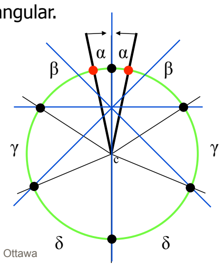
Problem: Case Switches

The algorithm:
If strict multiplicity: move there
Else If equiangular: move to center
Else elect some robots to gather



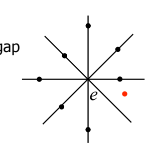
Cautious Movements

The moving robots stop at any critical point where the configuration becomes equiangular.



Finding Critical Points

- 1 starting position:
  is there center of equiangularity with 1 gap



- 2 starting positions:
  same with 2 gaps
- 3..n-1 starting positions:
  cannot generate equiangular configurations

Other Studies on Gathering

- Cieliebak, LATIN 2004
  - Non-oblivious robots, ASYNC
  - No multiplicity detection
- Gathering via Center of Gravity
  - Peleg et al.
- Gathering with faulty robots

Other Studies on Gathering

- Gathering with dynamic compasses
  - Katayama et al., SemiSync
- Gathering with limited visibility
  - Flocchini et al., TCS 2005, ASYNC

Problem 1.a: Gathering via CoG

Peleg et al.

Center-of-Gravity (CoG) algorithms

General approach:
In each cycle, the robots:
1. calculate some median position of the group
2. move towards that position

Natural variant:
Use the Center-of-Gravity (CoG)
(a.k.a. center of mass / barycenter)
of the robot group $\vec{c}[t] = \frac{1}{N}\sum_{i=1}^{N} \vec{r}_i[t]$

Note

- The analysis is performed in a d-dimensional space
- Easy consequence: convergence for $n=2$ robots using CoG algorithm

Analyzing CoG algorithm

In fully synchronous model: easy to analyze

In semi-synchronous / asynchronous model (more involved):

- Robots take measurements at different times, including while other robots are in movement
- Might cause oscillatory effects on the CoGs calculated by the various robots
- Might cause robots to pass each other by

Advantages of CoG algorithm

- Requires simple & efficient calculation - simple hardware, low computational effort.
- Applies to 1/2/3 dimensions and any # of robots.
- Bounded and simple-to-calculate rounding error
- Oblivious (i.e., requires no memory of previous actions & positions), hence
  - Memory-efficient
  - Self-stabilizing (i.e., finite number of transient errors cannot prevent eventual convergence).
- Prevents deadlocks (i.e., every robot can move at any given position unless already at center of gravity)

August 15, 2010 MAC 2010, Ottawa

Results

- CoG algorithm is correct in both the semi-synchronous and asynchronous model, for any # of robots

August 15, 2010 MAC 2010, Ottawa

Notation

Assume: Robots reside on x-axis.
(Handle each axis separately)

$r_i[t]$ = location of robot i at time t

$c_i[t]$ = CoG last calculated by robot i prior to time t

$c[t]$ = True CoG at time t

$H[t]$ = convex hull (smallest containing interval) of all $r_i[t]$ & $c_i[t]$

August 15, 2010 MAC 2010, Ottawa

Monotonicity of convex hull

Lemma: $H[t]$ cannot increase in time.

Lemma: In ASYNC, in any execution of CoG algorithm, over any $O(n^2)$ time interval, the convex hull of robot locations + CoGs is halved (in each dimension separately).

August 15, 2010 MAC 2010, Ottawa

Semi-synchronous model

Lemma:

In any execution of CoG algorithm, over any $O(n)$ time interval, the convex hull of robots + CoGs is halved (in each dimension separately).

August 15, 2010 MAC 2010, Ottawa

Fully synchronous model

Lemma: In any execution of CoG algorithm, robots achieve gathering in $4\ h[0]\ d^{3/2}$ time.

d: #dimensions
h[0]: max width of H[0]

August 15, 2010 MAC 2010, Ottawa

Problem 1.b: Self-stabilizing Deterministic Gathering

Petit et al., ALGOSENSORS 2009

We know that....

*All solutions so far assume that the robots start from **distinct** positions*

That is, even if the robots are oblivious, the solutions are not truly self-stabilizing

Petit et al. presented a deterministic protocol for solving the self-stabilizing gathering

- SSYNC
- Multiplicity Detection

We know that....

We know that the problem is unsolvable for 2 robots in SSYNC; hence, it is also unsolvable for an even number or robots

- The robots occupy at the beginning only two distinct positions on the plane, with n/2 robots on each position

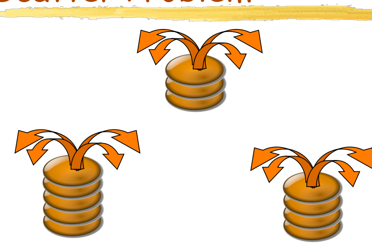
We know that....

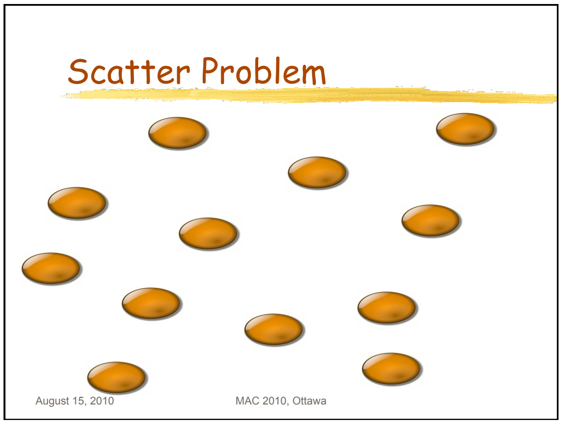
Petit et al presented a deterministic protocol for solving the (self-stabilizing) gathering for an odd number of robots, starting from any configuration

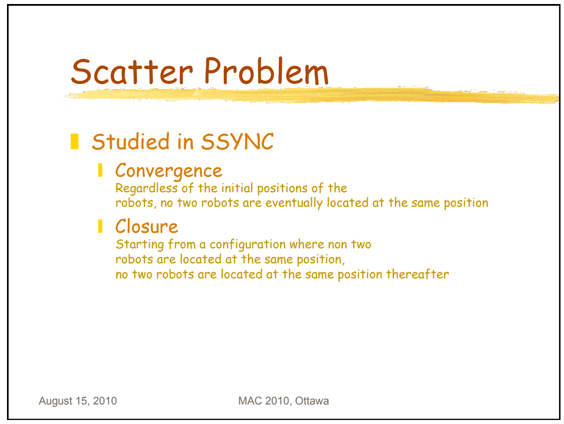
- SSYNC
- Strong Multiplicity Detection

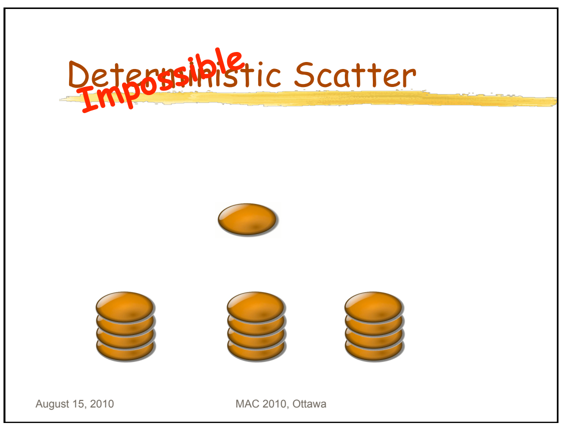
Problem 2: Scattering Petit et. al., FUN 2007

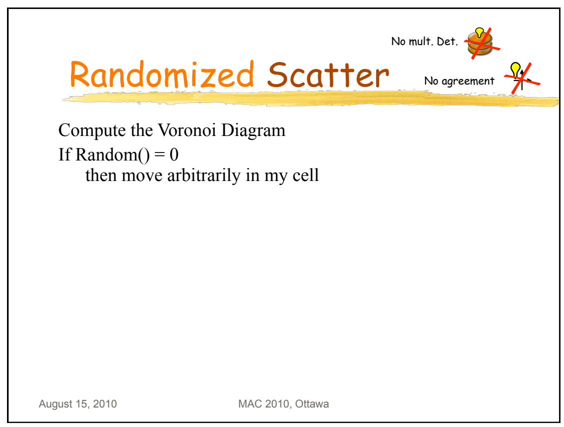
Scatter Problem

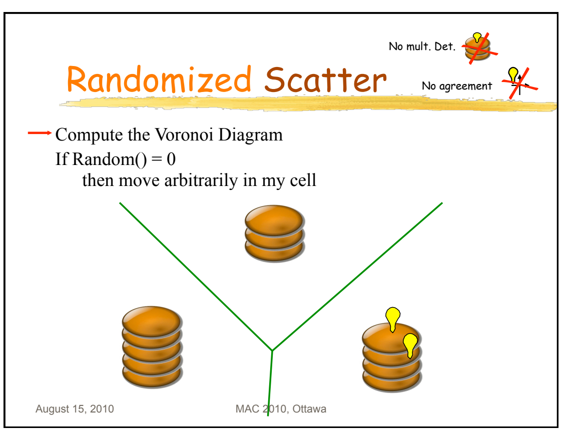


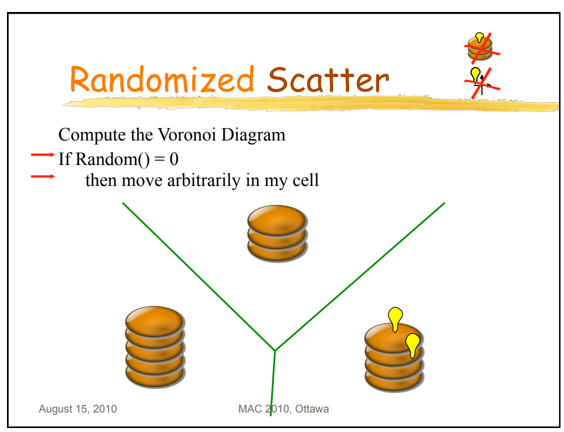

Scatter Problem
August 15, 2010
MAC 2010, Ottawa


Scatter Problem
Studied in SSYNC
Convergence
Regardless of the initial positions of the robots, no two robots are eventually located at the same position
Closure
Starting from a configuration where non two robots are located at the same position, no two robots are located at the same position thereafter
August 15, 2010
MAC 2010, Ottawa


Deterministic Scatter
Impossible
August 15, 2010
MAC 2010, Ottawa

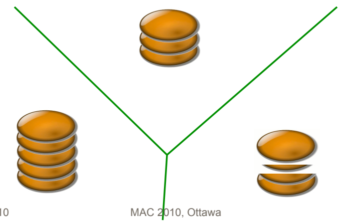

No mult. Det.
Randomized Scatter
No agreement
Compute the Voronoi Diagram
If Random() = 0
then move arbitrarily in my cell
August 15, 2010
MAC 2010, Ottawa


No mult. Det.
Randomized Scatter
No agreement
Compute the Voronoi Diagram
If Random() = 0
then move arbitrarily in my cell
August 15, 2010
MAC 2010, Ottawa


Randomized Scatter
Compute the Voronoi Diagram
If Random() = 0
then move arbitrarily in my cell
August 15, 2010
MAC 2010, Ottawa

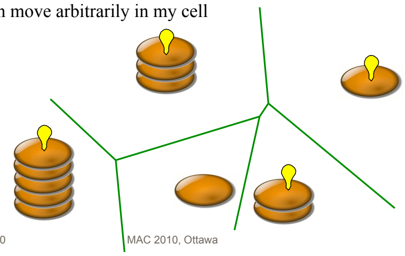
Randomized Scatter

Compute the Voronoi Diagram
If Random() = 0
then move arbitrarily in my cell



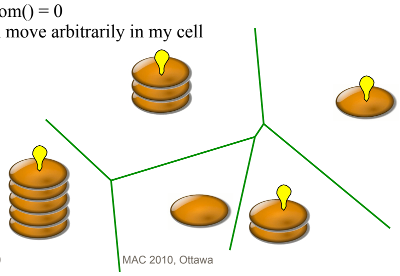
Randomized Scatter

Compute the Voronoi Diagram
If Random() = 0
then move arbitrarily in my cell



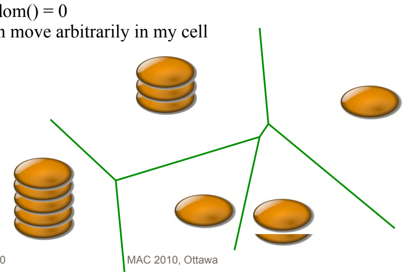
Randomized Scatter

Compute the Voronoi Diagram
If Random() = 0
then move arbitrarily in my cell



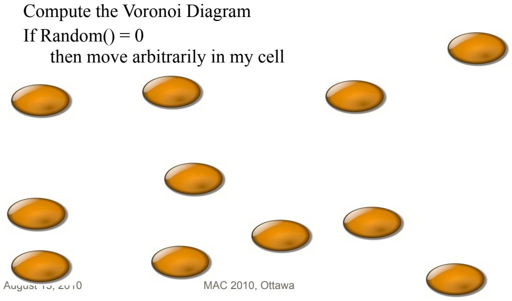
Randomized Scatter

Compute the Voronoi Diagram
If Random() = 0
then move arbitrarily in my cell



Randomized Scatter

Compute the Voronoi Diagram
If Random() = 0
then move arbitrarily in my cell



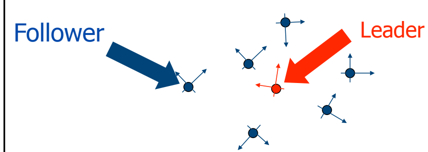
Randomized Scatter

- It could be used as a starting procedure for using previous gathering or arbitrary pattern formation
  - Multiplicity detection is necessary to switch
  - Eliminates the initial condition of having "distinct position"
  - Randomized Self-Stabilizing Solutions

Problem 3: Flocking

August 15, 2010 MAC 2010, Ottawa

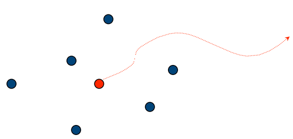
Flocking
Unlimited Visibility



Followers recognize Leader
No Agreement on Local Axes

August 15, 2010 MAC 2010, Ottawa

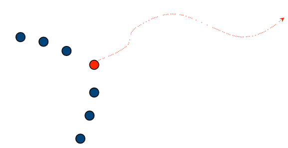
Flocking
Unlimited Visibility



Leader acts independently (e.g., human driven)

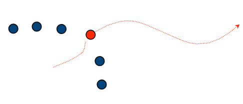
August 15, 2010 MAC 2010, Ottawa

Flocking
Unlimited Visibility



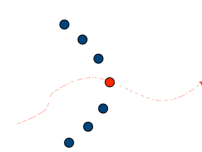
August 15, 2010 MAC 2010, Ottawa

Flocking
Unlimited Visibility



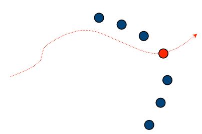
August 15, 2010 MAC 2010, Ottawa

Flocking
Unlimited Visibility



August 15, 2010 MAC 2010, Ottawa

Flocking
Unlimited Visibility



Assumption

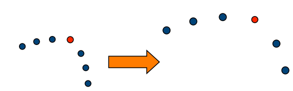
Time spent in *Look* and *Compute* is negligible w.r.t. the time spent in *Move*

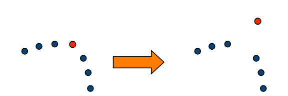
The constraints

- **No** shared coordinate system
- **No** way to observe direction of movement
- **No** common velocity
- **Common** pattern
  - In order for the problem to be significant!

Necessary conditions

- Common distance unit
  - Otherwise, the followers could scale the pattern instead of following the leader
- Fast enough
  - The leader cannot be faster than the followers

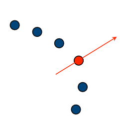




The Solution (Dam 2004)

Followers do not know the path to follow in advance

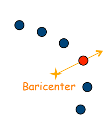
The algorithm lets Followers only form patterns that are symmetric w.r.t. direction of movement of the Leader



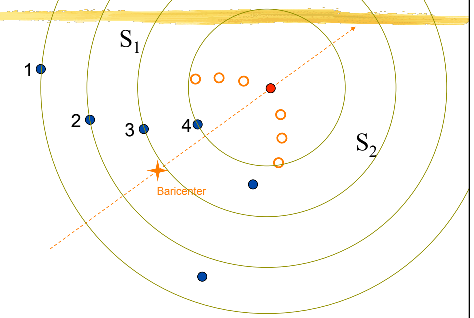
The Solution (Dam 2004)

Followers need a way to approximate the direction of movement of the Leader

Estimated direction of movement: Given by axis passing through baricenter of Followers' positions and Leader's position

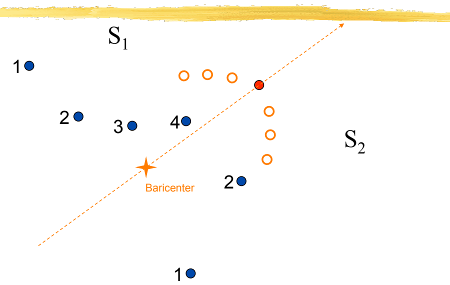


The Algorithm (Dam 2004)



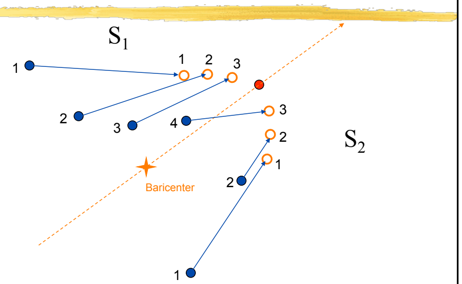
August 15, 2010 MAC 2010, Ottawa

The Algorithm (Dam 2004)



August 15, 2010 MAC 2010, Ottawa

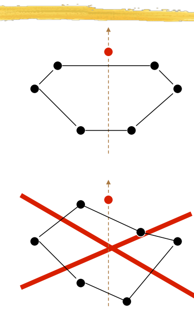
The Algorithm (Dam 2004)



August 15, 2010 MAC 2010, Ottawa

Fleet symmetry

- The pattern must be symmetric wrt the direction of movement of the leader
- Other patterns may be formed with more agreement (e.g., 1 axis)
  - Not studied yet!!



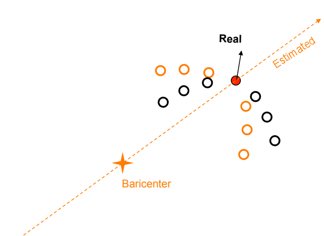
August 15, 2010 MAC 2010, Ottawa

A few quirks

- Stability: ships may have to give way
- Limits on the angular velocity of the leader
- Real heading of the leader (and thus formation) different from estimated

August 15, 2010 MAC 2010, Ottawa

Real vs estimated heading



August 15, 2010 MAC 2010, Ottawa

Measures

- Given a set of ships F and a set of points Z (the pattern), we define:

$$\Delta_{F,Z}(t) = \min_{\pi \in \Pi} \sum_{i=1}^{|F|} d(f_i(t), z_{\pi_i}(t))$$

- Experimental measures:
  - Distance from estimated formation (Δ_e)
  - Distance from real formation (Δ_r)

Experimental results

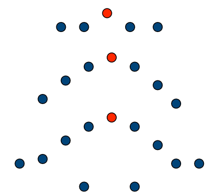
Tests of the solution with four formations:

Line (4)

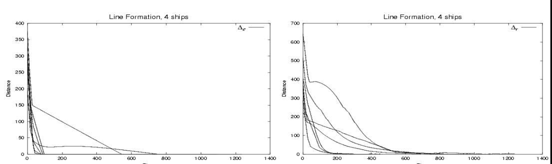
Wedge (6)

Spread (10)

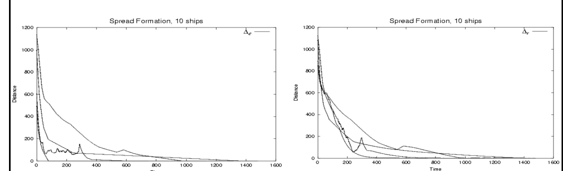
Random (2-8)



Experimental results



Experimental results



Experimental results

Leader

Problem 4: Intruder

The intruder problem

- An **intruder** is trying to sneak through a restricted area
- A number of autonomous units (robotic **cops**) is patrolling the area
- The cops have to **surround** the intruder


The intruder problem

- Example: sensible infrastructures
  - Airfield runway
  - Logistic compound
  - Dam or power plant
- Example: hostile area
  - Battlefield
  - Police operation

The cops

- Units must be truly **autonomous**
  - Completely asynchronous
  - Undistinguishable (i.e., no IDs)
  - No explicit communications (i.e., no radio)
  - No shared knowledge
    - No common compass
    - No common chirality

The cops

- Units must be truly **autonomous**
  - Completely asynchronous
  - Undistinguishable (i.e., no IDs)
  - No explicit communications (i.e., no radio)
  - No shared knowledge
    - No common compass
    - No common chirality
- Algorithm must work with a **variable number** of cops
  - Cops could be knocked-off by the intruder

Computational model

- We consider an infinite plane
- Both the intruder and the cops are modeled as units that can freely move on the plane

- All cops move according to a given **algorithm** (the same for all the units)
- The intruder moves **independently** from other units; its course is not known in advance

Computational model

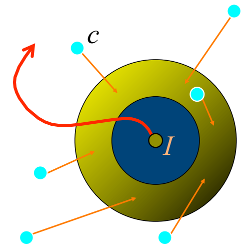
- Cops have **sensors** that report the **positions** of the intruder and of other cops
- Cops are **oblivious** – they have no memory, and do not rely on stored information about the past

Computational model

- As in the flocking, we assume that
  - Time spent in *Look* and *Compute* is negligible w.r.t. the time spent in *Move*

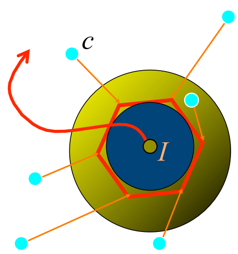
Formalizing the problem

- The cops must occupy a **ring** around the intruder:
  $r_1 \le \mathrm{dist}(c, I) \le r_2$
- The cops must be evenly spaced, minimazing the maximal "**escape hole**" in the ring

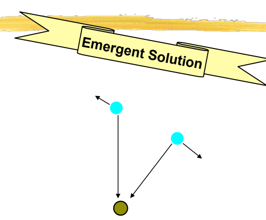


Formalizing the problem

- Both conditions can be met by forming a regular n-gon of characteristic angle $2\pi/n$ and radius r_1



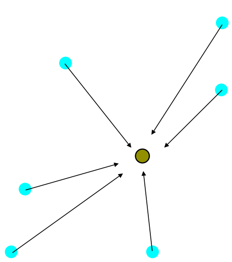
Two Algorithms: Heuristic (FUN 2004)



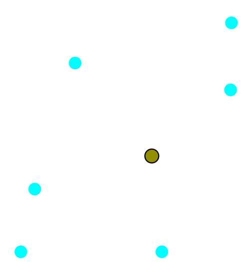
Heuristic approach

- All robots are subject to two forces:
  - Attractive, towards the enemy
  - Repulsive, mutual
- Parameters are tuned so that the aequilibrium is a solution for the problem

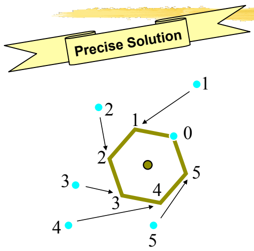
The HEUR-S Algorithm



The HEUR-S Algorithm



Two Algorithms: LAT



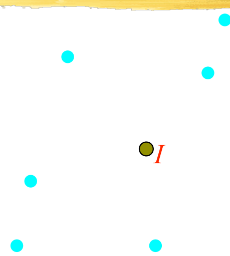
Algorithmic approach

- The robots reach an agreement on their ordering
- Each one establishes its own target
- They reach the target (taking care not to overtake each other)

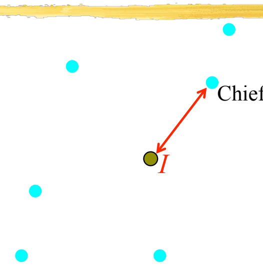
The LAT algorithm (idea)

- The closest cop to the intruder is identified and declared *Chief*

The LAT algorithm (idea)



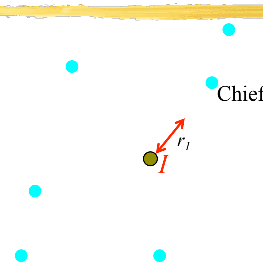
The LAT algorithm (idea)



The LAT algorithm (idea)

- The closest cop to the intruder is identified and declared *Chief*
- The Chief moves towards the intruder, up to a distance r_1

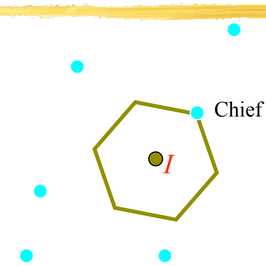
The LAT algorithm (idea)



The LAT algorithm (idea)

- The closest cop to the intruder is identified and declared *Chief*
- The Chief moves towards the intruder, up to a distance r_1
- The other cops scale and align the n-gon so that the center is on I and a vertex on the Chief

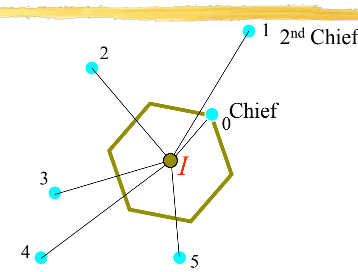
The LAT algorithm (idea)



The LAT algorithm (idea)

- The closest cop to the intruder is identified and declared *Chief*
- The Chief moves towards the intruder, up to a distance r_1
- The other cops scale and align the n-gon so that the center is on I and a vertex on the Chief
- All cops are lexicographically sorted according to their angle with I-Chief
  - Orientation is Chief → 2nd Chief

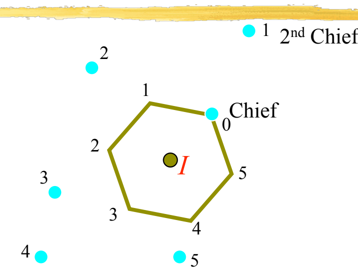
The LAT algorithm (idea)



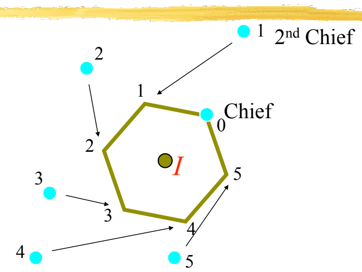
The LAT algorithm (idea)

- The closest cop to the intruder is identified and declared *Chief*
- The Chief moves towards the intruder, up to a distance r_1
- The other cops scale and align the n-gon so that the center is on I and a vertex on the Chief
- All cops are lexicographically sorted according to their angle with I-Chief
  - Orientation is Chief → 2nd Chief
- The i-th cop in the ordering moves (up to ε) toward the i-th vertex of the scaled and aligned n-gon

The LAT algorithm (idea)



The LAT algorithm (idea)



August 15, 2010 MAC 2010, Ottawa

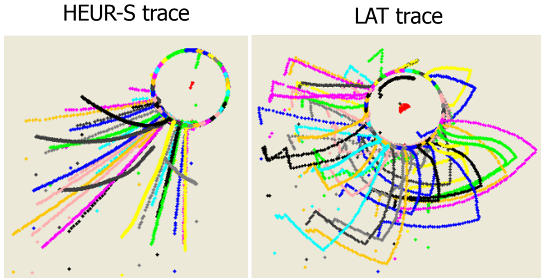
Some pitfalls….

- Multiple candidate chiefs
  - All cops wait for the intruder to move, breaking the tie
- A cop may have to go to the opposite side of the ring
  - Perform a *sideway step* to avoid a change of Chief

August 15, 2010 MAC 2010, Ottawa

Two Algorithms

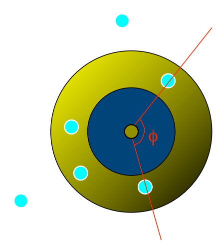
HEUR-S trace LAT trace



August 15, 2010 MAC 2010, Ottawa

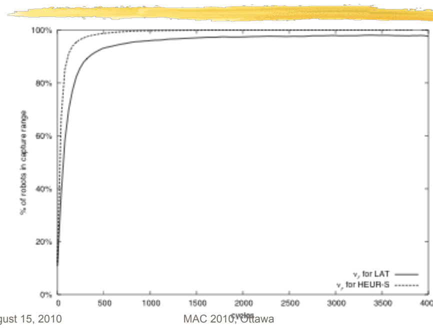
Evaluation of the algorithm

- Numerical simulations
- Two kind of measures:
  - What fraction of the cops is inside the target ring?
  - How large is the largest hole in the ring, w.r.t. the optimal hole?



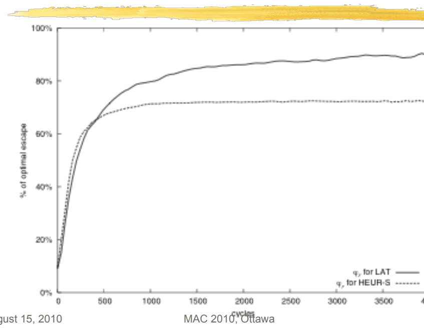
August 15, 2010 MAC 2010, Ottawa

Fraction of Robots in the Capture Area



August 15, 2010 MAC 2010, Ottawa

Size of Largest Hole (relative)



August 15, 2010 MAC 2010, Ottawa

Adaptiveness

- The algorithms adapts to dynamic changes in the number of cops
  - Cops can join
  - Cops can leave

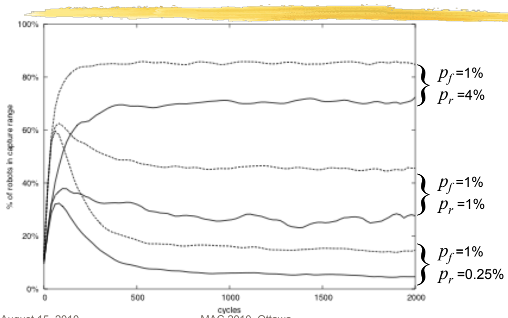
August 15, 2010 MAC 2010, Ottawa

Fault Tolerance

- Robots could be blocked, knocked off by the enemy, or otherwise suffer **faults**
- We consider **transient** faults
  - A faulty robot stops moving, but continues sensing and computing
  - Fault model based on two parameters:
    - p_f – probability of occurrence of a fault
    - p_r – probability of resuming normal behaviour
- Faults occur indefinitely

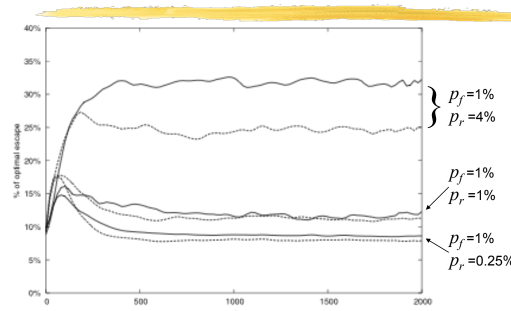
August 15, 2010 MAC 2010, Ottawa

Robots in the Capture Area (in the presence of faults)



August 15, 2010 MAC 2010, Ottawa

Size of the Largest Hole (in the presence of faults)

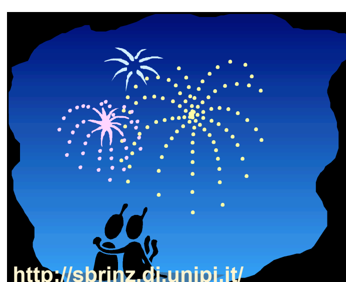


August 15, 2010 MAC 2010, Ottawa

Summary

- Gathering
  - ASYNC
  - CoG (ASYNC and SSYNC)
  - Self-Stabilizing Gathering in SSYNC
- Scattering (SSYNC)
- Flocking
- Intruder

August 15, 2010 MAC 2010, Ottawa



Thanks

M. Cieliebak – Switzerland
P. Flocchini – Canada
V. Gervasi – Italy
N. Santoro – Canada
P. Widmayer – Switzerland

THE END

August 15, 2010 MAC 2010, Ottawa

References

1. Efrima and D. Peleg, Distributed Algorithms for Partitioning A Swarm of Autonomous Mobile Robots, 14th SIROCCO, June 2007, Italy.
2. A. Efrima and D. Peleg, Distributed Models and Algorithms for Mobile Robot Systems, 33rd SOFSEM, Jan. 2007, Czech Republic, LNCS 4362, 70-87.
3. R. Cohen and D. Peleg, Local Algorithms for Autonomous Robot Systems, SIROCCO 13th, LNCS 4056, Chester, England, June 2006, 29-43.
4. R. Cohen and D. Peleg, Convergence of Autonomous Mobile Robots With Inaccurate Sensors and Movements, 23rd STACS, Marseille, France, Feb. 2006, 549-560. (Also as Research Report MCS04-08, the Weizmann Institute of Science, 2004.)
5. D. Peleg, Distributed Coordination Algorithms for Mobile Robot Swarms: New Directions and Challenges, IWDC 7th, Dec. 2005, Kharagpur, India, 1-12.
6. R. Cohen and D. Peleg, Convergence Properties of the Gravitational Algorithm in Asynchronous Robot Systems, 12th ESA, Bergen, Norway, Sept. 2004, 228-239.
7. R. Cohen and D. Peleg, Robot Convergence via Center-of-Gravity Algorithms, 11th SIROCCO, Bratislava, Slovakia, June 2004, 79-88.
8. N. Agmon and D. Peleg, Fault-Tolerant Gathering Algorithms for Autonomous Mobile Robots, 15th SODA, New Orleans, Louisiana, January 2004.
9. Y Dieudonné and F Petit. Circle Formation of Weak Robots and Lyndon Words.Information Processing Letters, vol. 104(4), pages 156-162, 2007.
10. Y Dieudonné and F Petit. Robots and Demons (The Code of the Origins). Fourth International Conference on Fun with Algorithms (FUN 2007), pages 108-119, LNCS 4475, 2007.

11. Y Dieudonné and F Petit. Swing Words to Make Circle Formation Quiescent. 14th International Colloquium on Structural Information and Communication Complexity (SIROCCO 2007), pages 162-175, LNCS 4474, 2007.
13. Y Dieudonné, O Labbani-Igbida, and F Petit. Circle Formation of Weak Mobile Robots. Eighth International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'06), pages 262-275, LNCS 4280, 2006.
14. H. Ando, Y. Oasa, I. Suzuki and M. Yamashita, ``Distributed memoryless point convergence algorithm for mobile robots with limited visibility,'' IEEE Transactions on Robotics and Automation, Vol. 15, No. 5, October 1999, pp. 818-828.
15. I. Suzuki and M. Yamashita, ``Distributed anonymous mobile robots - formation of geometric patterns,'' SIAM Journal on Computing, Vol. 28, No. 4, 1999, pp. 1347-1363.
16. I. Suzuki and M. Yamashita, ``Agreement on a common x-y coordinate system by a group of mobile robots'' in Intelligent Robots: Sensing, Modeling and Planning, Bolles, Bunke and Noltemeier Eds., World Scientific, 1997, pp. 305-321.
17. Y. Oasa, I. Suzuki and M. Yamashita, ``A robust distributed convergence algorithm for autonomous mobile robots,'' in Proceedings of the 1997 IEEE International Conference on Systems, Man and Cybernetics, Orlando, Florida, pp. 287-292, October 1997.
18. I. Suzuki and M. Yamashita, ``Distributed Anonymous Mobile Robots-Formation and Agreement Problems,'' in Proceedings of the 3rd International Colloquium on Structural Information and Communication Complexity (SIROCCO '96), Siena, Italy, June 1996, pp. 313-330.
19. K. Sugihara and I. Suzuki, ``Distributed algorithms for formation of geometric patterns with many mobile robots,'' Journal of Robotic Systems, Vol. 13, No. 3, March 1996, pp. 127-139.

19. H. Ando, I. Suzuki and M. Yamashita, ``Formation and agreement problems for synchronous mobile robots with limited visibility,'' Proceedings of the 1995 IEEE International Symposium on Intelligent Control, August 1995, pp. 453-460.
20. I. Suzuki and M. Yamashita, ``Formation and agreement problems for anonymous mobile robots,'' in Proceedings of the 31st Annual Allerton Conference on Communication, Control, and Computing, University of Illinois, Urbana, Illinois, October 1993, pp. 93-102.
23. Samia Souissi, Xavier Défago and Masafumi Yamashita. Gathering asynchronous mobile robots with inaccurate compasses. In A. Shvartsman, editor, Proc. 10th Intl. Conf. on Principles of Distributed Systems (OPODIS 2006), volume 4305 of LNCS, pages 333––349, Bordeaux, France, December 2006. Springer.
24. Samia Souissi, Xavier Défago and Masafumi Yamashita. Using eventually consistent compasses to gather oblivious mobile robots with limited visibility. In A. K. Datta and M. Gradinariu, editors, Proc. 8th Intl. Symp. on Stabilization, Safety, and Security of Distributed Systems (SSS 2006), volume 4280 of LNCS, pages 471––487, Dallas, TX, USA, November 2006. Springer.
25. Xavier Défago, Maria Gradinariu, Stéphane Messika and Philippe Raipin-Parvédy. Fault-tolerant and self-stabilizing mobile robots gathering. In S. Dolev, editor, Proc. 20th Intl. Symp. on Distributed Computing (DISC 2006), volume 4167 of LNCS, pages 46––60, Stockholm, Sweden, September 2006. Springer.
26. Xavier Défago and Akihiko Konagaya. Circle formation for oblivious anonymous mobile robots with no common sense of orientation. In Proc. 2nd Int'l Workshop on Principles of Mobile Computing (POMC'02), pages 97–104, Toulouse, France, October 2002. ACM.
27. Xavier Défago. Distributed computing on the move: From mobile computing to cooperative robotics and nanorobotics (a position paper). In Proc. Int'l Workshop on Principles of Mobile Computing (POMC'01), pages 49–55, Newport, RI, USA, August 2001. ACM.
28. http://sbrinz.di.unipi.it/~peppe/Publications