Almost optimal asynchronous rendezvous in infinite multidimensional grids

Evangelos Bampas[1] Jurek Czyzowicz[2]
Leszek Gąsieniec[3] David Ilcinkas[1] Arnaud Labourel[1]

[1]LaBRI: INRIA Bordeaux, CNRS, and University of Bordeaux 1

[2]University of Québec

[3]University of Liverpool

DISC 2010

The (grid) rendezvous problem

The problem (in the grid)

Two mobile agents must meet in a grid.

- Terrain → grid of dimension 2
- Mobile agents → points moving from node to node along the edges choosing at each step a direction (N,S,W,E)
- Rendezvous → meeting of the two agents on a node or in an edge
- Cost → sum of the lengths of the trajectories of the agents until rendezvous

Asynchronous model

The agents

The agents try to choose their routes so they always meet.

The omniscient adversary

- Tries to prevent the rendezvous.
- Knows in advance the route chosen by the agent (rendezvous algorithm).
- Chooses the starting positions of the agents.
- Determines the speed of each agent at any time on its route (the speed can be 0 but only for a finite amount of time).

Related work

[1] Czyzowicz, Labourel, and Pelc, *How to meet asynchronously (almost) everywhere*, SODA 2010.

- Asynchronous rendezvous is feasible in (almost) any unknown, anonymous graph when the agents know only their identities.

[2] Collins, Czyzowicz, Gąsieniec, Labourel, *Tell me where I am so I can meet you sooner*, ICALP 2010.

- Asynchronous rendezvous with location information in grids with cost $O(D^{2+\epsilon})$.

D: distance between the starting positions of the agents

Rendezvous in grids with total knowledge

Claim

There is a rendezvous algorithm at cost D if each of the agents knows its starting position and the starting position of the other agent.

D: distance between the two starting positions of the agents

Rendezvous in grids with no knowledge

Theorem [1]

There is a rendezvous algorithm if the agents do not know their starting positions but have distinct identities. The cost of the rendezvous is exponential in D and in the identities of the agents.

D: distance between the two starting positions of the agents
Identities : binary words needed to break symmetry in the grid for a deterministic algorithm. The agents must meet for any pair of identities chosen by the adversary.

This work: partial knowledge

Rendezvous in the grid with partial knowledge

There is a rendezvous algorithm at cost $O(D^2 \log^7 D)$ if each agent knows its initial position (same system of coordinates for both agents).

Almost optimal since there is a lower bound of $\Omega(D^2)$.

Lower bound of $\Omega(D^2)$

- The D-neighborhood of any node contains $\Theta(D^2)$ nodes.
- The adversary may stall one of the agents arbitrarily long at its starting position.
- Therefore, the other agent eventually has to explore its D-neighborhood.

Generalization to higher dimensions

Rendezvous in the grid of dimension δ

There is a rendezvous algorithm at cost $O(D^\delta \log^{\delta^2+\delta+1} D)$ if each agent knows its initial position (same system of coordinates for both agents).

Almost optimal since there is a lower bound of $\Omega(D^\delta)$.

Erroneous strategy 1

Go to O

Go to the origin and wait for the other agent.

Problem: the cost depends on the distance of the agents from the origin and not on the distance between their initial positions (no match with the lower bound).

Erroneous strategy 2

Use algorithm for the line

Construct a simple space filling curve in the grid and use a known polynomial-cost algorithm on the line to achieve rendezvous.

Problem: for any simple space-filling curve, there exists a pair of close points in the plane, such that their distance along the space-filling curve is arbitrarily large [3].

[3] Gotsman and Lindenbaum, *On the metric properties of discrete space-filling curves*, IEEE Transactions on Image Processing, 5(5), pp. 794-797, 1996.

The right strategy

Space-covering sequence★

Construct a space-covering sequence (non-simple curve) covering the infinite grid.

- Recursive construction using an infinite hierarchy of partitions (levels) of the grid into squares of increasing sizes.

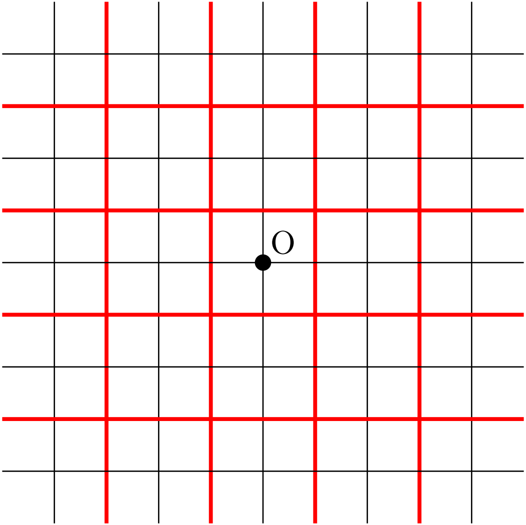
★ introduced in [2]

The hierarchy of partitions $\mathcal{C}$

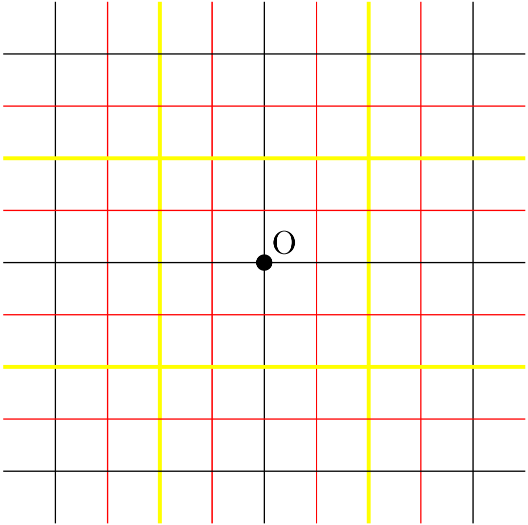
Hierarchy $\mathcal{C}$

Central-square hierarchy $\mathcal{C}$: centered square partition.
C_i : partition into squares of side length 2^i with the origin at the center of a square ($\forall i \geq 1$).

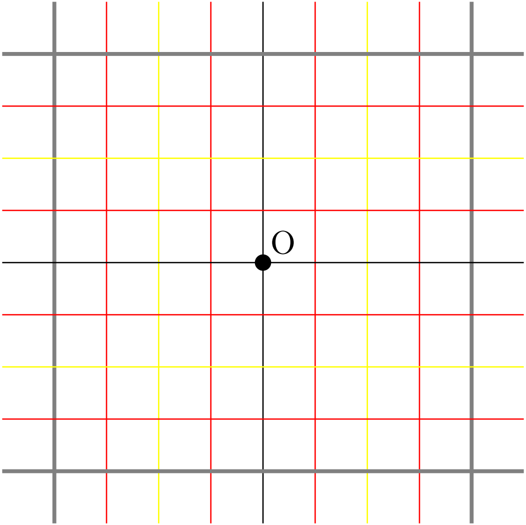
The first level of $\mathcal{C}$: C_1



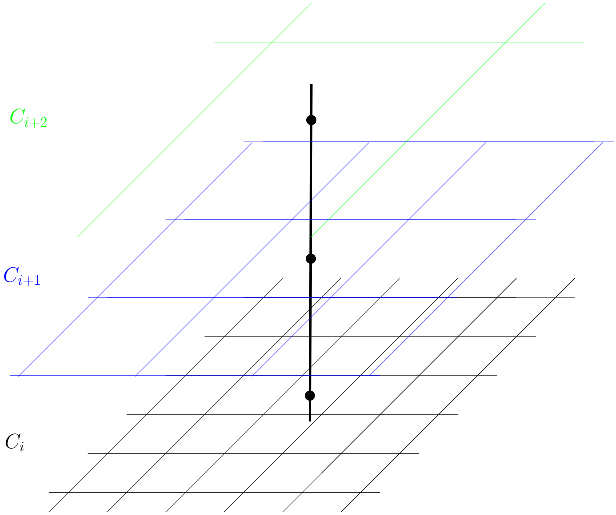
The second level of $\mathcal{C}$: C_2



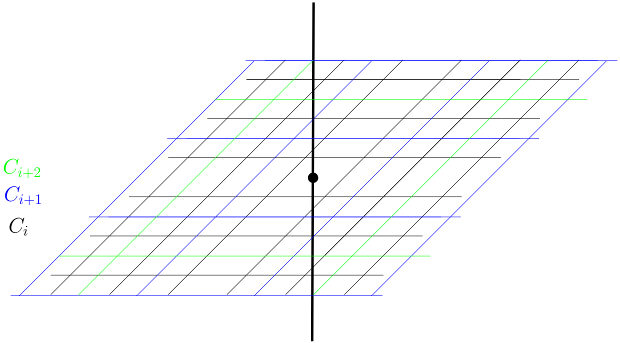
The third level of $\mathcal{C}$: C_3



Levels in $\mathcal{C}$



Levels in $\mathcal{C}$



Tree-like structure $\mathcal{C}$

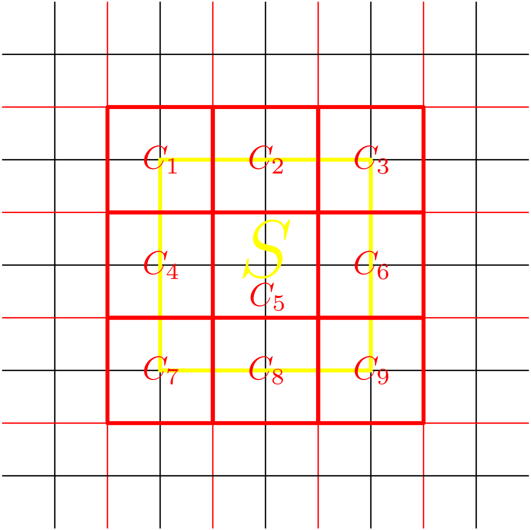
Tree-like structure

A square in level C_i is a child of a square in level C_{i+1} if their intersection is non-empty.

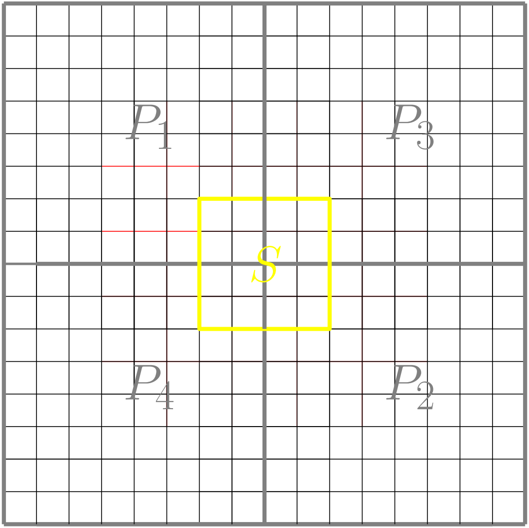
Remark 1: a square has 9 children.

Remark 2: a square can be the child of multiple squares (at most four).

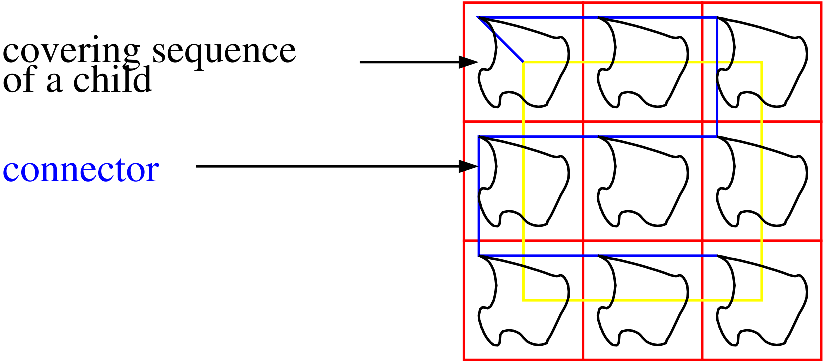
Children of a square S



Parents of a square S



Covering sequence of a square

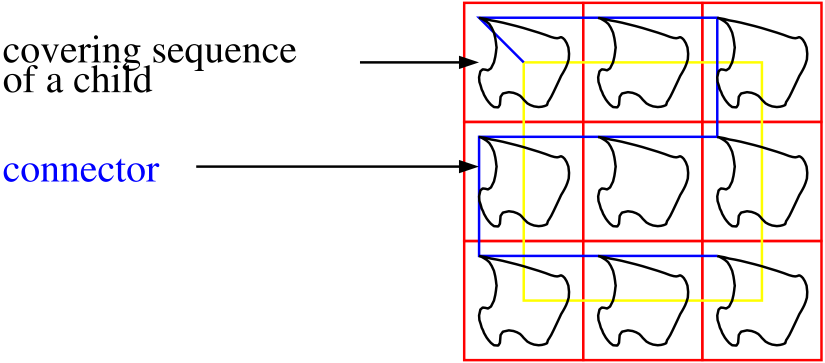


Rendezvous algorithm

0. Cover the starting square of C_1

1. Having just covered square S, go to its parent P by following backwards the covering sequence of P
2. Cover P
3. Repeat from 1

Rendezvous algorithm



Final steps

Size of rendezvous square

For any two points at distance D and any three consecutive partitions of size at least $4D$, there exists a square of one of the three partitions that contains both points.

Skipping levels

Instead of using all levels of $\mathcal{C}$, we use only levels i_j defined by:

$i_1 = 1$

$i_{j+1} = i_j + \max\{\lceil \log i_j \rceil, 1\}$

Result

Rendezvous in dimension δ

The rendezvous algorithm ensures rendezvous in the δ-dimensional grid with cost $O(D^{\delta} \log^{\delta^2+\delta+1} D)$.

Can we close the polylog gap?

Open problem: partial knowledge

Does there exist an asynchronous deterministic algorithm in the grid such that the cost of rendezvous is $\Theta(D^2)$ if each agent knows its initial position?

Generalization to graphs other than grids

Open problem: location information

Does there exist an asynchronous deterministic algorithm in any graph such that the cost of rendezvous is polynomial in D if each agent knows the graph and its initial position?

Thank you