

# Comment se rencontrer de manière asynchrone dans un graphe quelconque?

Jurek Czyzowicz<sup>1</sup>, Arnaud Labourel<sup>2</sup>, Andrzej Pelc<sup>1</sup>

<sup>1</sup>Département d'informatique, Université du Québec en Outaouais,  
Gatineau, Québec, Canada

<sup>2</sup>LaBRI, Université Bordeaux, Talence, France

11<sup>ème</sup> Journées Graphes et Algorithmes (JGA 2009)  
5&6 novembre 2009

# Le problème du rendez-vous

## Le problème

Deux agents mobiles doivent se rencontrer dans un réseau inconnu.

Modélisation du problème :

- Réseaux  $\rightarrow$  graphes
- Agents mobiles  $\rightarrow$  points se déplaçant de sommets en sommets le long des arêtes du graphe

Complexité :

longueur des routes des agents jusqu'à la rencontre

# Rendez-vous dans les graphes

## Graphes considérés

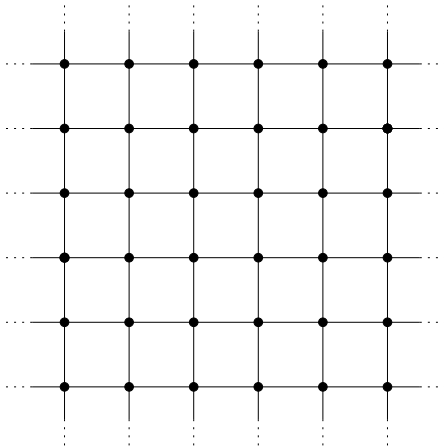
- Graphes énumérables (ensemble de sommet énumérable fini ou infini)
- Sommets anonymes
- Arêtes incidentes à un sommet numéroté localement (numéro de port)

## Déplacement des agents

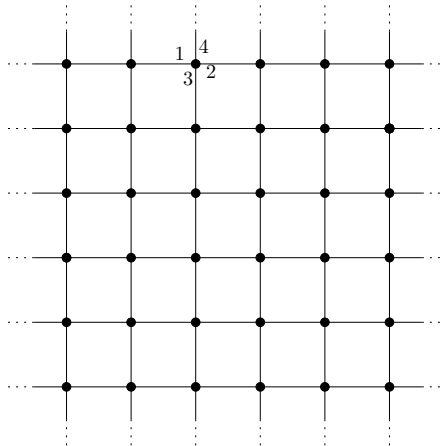
À chaque étape :

- choix d'un numéro de port sur le sommet courant
- déplacement le long de l'arête correspondante
- récupération du numéro de port de l'arête dans le nouveau sommet (numéro de port sortant)

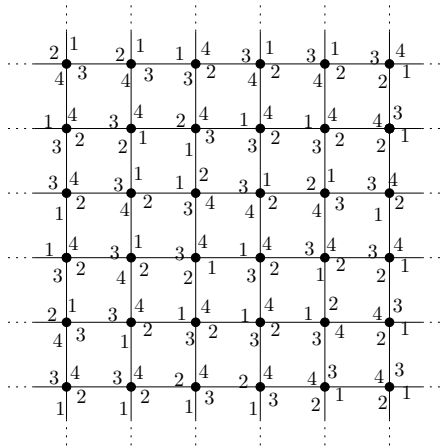
# Exemple de graphe



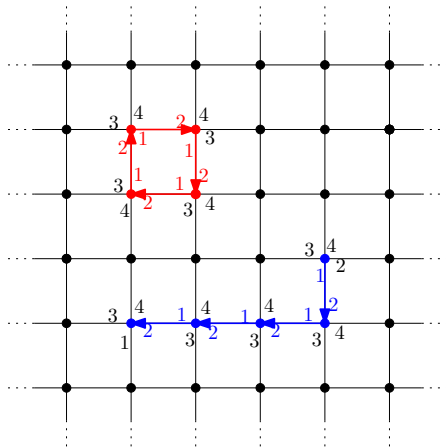
# Exemple de graphe



# Exemple de graphe



# Conséquence du fait que les sommets sont anonymes



# Rendez-vous déterministe

## Algorithme déterministe

La route (séquence des numéro de ports) empruntée par un agent ne dépends que :

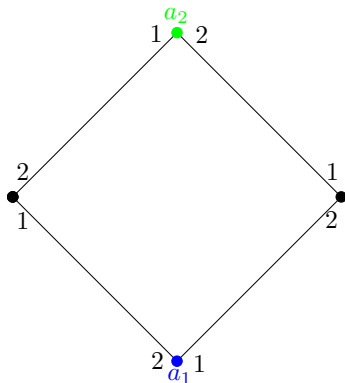
- du graphe et de la position de départ de l'agent
- de l'identifiant de l'agent

Identifiant indispensable dans le modèle déterministe

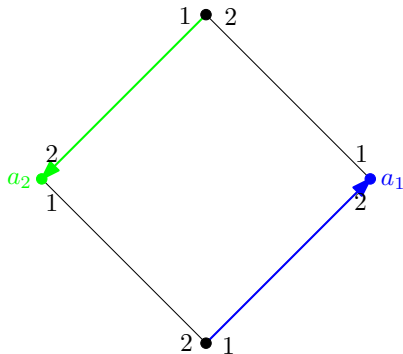
⇒ Sans identifiant, le rendez-vous est impossible dans un cycle à cause de la symétrie.



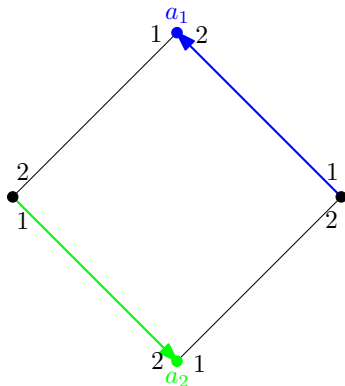
# Rendez-vous déterministe impossible sans identifiant



# Rendez-vous déterministe impossible sans identifiant



# Rendez-vous déterministe impossible sans identifiant



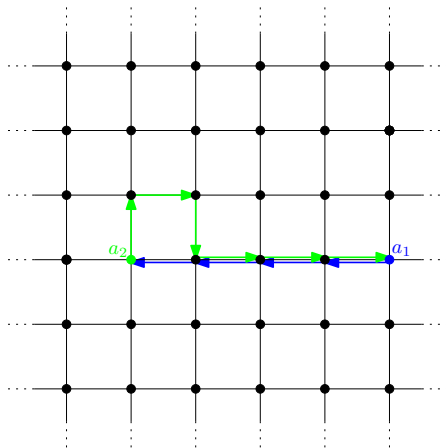
# Asynchronicité

## Modèle asynchrone

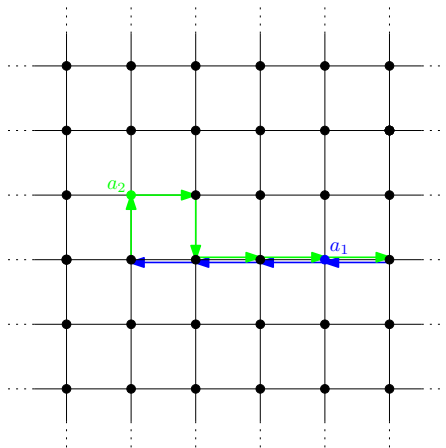
La durée des étapes de chaque agent est déterminée par un adversaire omniscient dont le but est d'empêcher le rendez-vous.

⇒ Il doit exister une correspondance entre les débuts des trajectoires des agents car sinon le rendez-vous peut être empêché par l'adversaire.

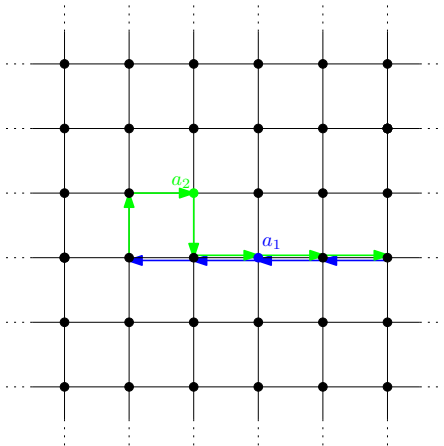
# Exemple de trajectoire n'assurant pas le rendez-vous



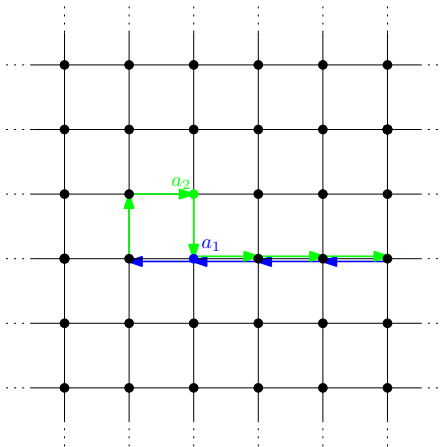
# Exemple de trajectoire n'assurant pas le rendez-vous



# Exemple de trajectoire n'assurant pas le rendez-vous

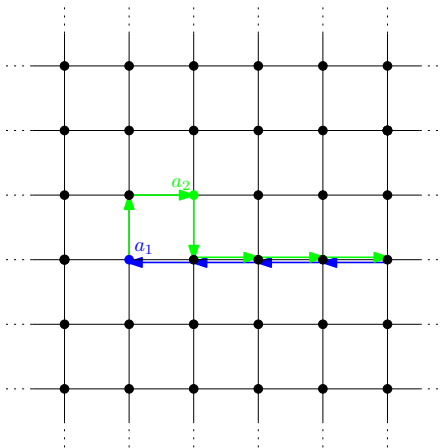


# Exemple de trajectoire n'assurant pas le rendez-vous

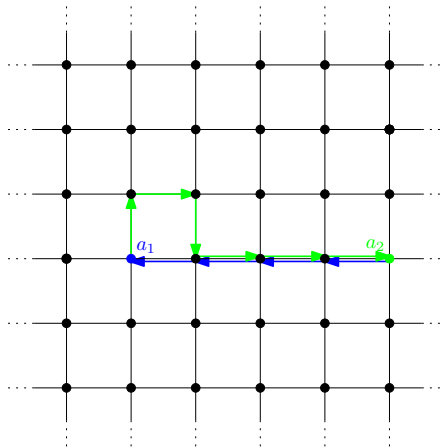




# Exemple de trajectoire n'assurant pas le rendez-vous



# Exemple de trajectoire n'assurant pas le rendez-vous



A diagram of a 2D square lattice. The lattice is represented by a grid of black dots (lattice points) connected by solid black lines. The grid is 6 units wide and 6 units high. The horizontal and vertical lines are labeled with ellipses at their ends, indicating they extend infinitely. Two basis vectors are shown:  $a_1$  is a blue vector pointing horizontally from a central point to the right, labeled with a blue  $a_1$  at its tip;  $a_2$  is a green vector pointing vertically from the same central point upwards, labeled with a green  $a_2$  at its tip. The vectors are represented by solid lines with arrowheads. Dashed lines of the same color (blue and green) form a rectangular path around the central point, with arrowheads indicating a counter-clockwise direction of travel.

# Chemin infini

## Rendez-vous dans le chemin infini [1,2]

Rendez-vous en  $O(D|L_{\min}|)$  quand  $D$  est connue et  
 $O(D \log^2 D + D \log D |L_{\max}| + D|L_{\min}|^2 + |L_{\max}||L_{\min}| \log |L_{\max}|)$   
sinon.

$|L_{\min}|, |L_{\max}|$  : taille en bits du plus petit et plus grand identifiant  
des agents respectivement.

[1] G. De Marco, L. Gargano, E. Kranakis, D. Krizanc, A. Pelc, U. Vaccaro, Asynchronous deterministic rendezvous in graphs, Theoretical Computer Science 355 (2006), 315-326.

[2] G. Stachowiak, Asynchronous Deterministic Rendezvous on the Line, SOFSEM 2009 : 497-508.

# Cycle

Rendez-vous dans le cycle de taille  $n$ [1]

Rendez-vous en  $O(n|L_{\min}|)$  quand  $n$  est connu et  $O(n|L_{\max}|)$  sinon.

$|L_{\min}|, |L_{\max}|$  : taille en bits du plus petit et plus grand identifiant des agents respectivement.

[1] G. De Marco, L. Gargano, E. Kranakis, D. Krizanc, A. Pelc, U. Vaccaro, Asynchronous deterministic rendezvous in graphs, Theoretical Computer Science 355 (2006), 315-326.

# Graphe fini quelconque

## Rendez-vous dans un graphe quelconque [1]

Rendez-vous possible (complexité exponentielle en la taille du graphe) si une borne sur la taille du graphe est connue.

[1] G. De Marco, L. Gargano, E. Kranakis, D. Krizanc, A. Pelc, U. Vaccaro, Asynchronous deterministic rendezvous in graphs, Theoretical Computer Science 355 (2006), 315-326.

# Rendez-vous dans les graphes

## Théorème

Le rendez-vous déterministe asynchrone est possible dans un graphe quelconque anonyme (fini ou infini).

# Idée générale de l'algorithme

## Principe de l'algorithme

Forcer le rendez-vous pour tous les configurations initiales possibles en créant un tunnel entre les routes des deux agents.

Tunnel : séquence d'arêtes  $e_1, e_2, \dots, e_k$  telle que :

- la route de l'agent 1 commence par  $e_1, e_2, \dots, e_k$
- la route de l'agent 2 commence par  $e_k, e_{k-1}, \dots, e_1$



# Fonctionnement de l'algorithme

**Configuration initiale :** quadruplet  $(L_1, S_1, L_2, S_2)$

- $S_1, S_2$  : suite fini d'entiers de même longueur.
- $L_1, L_2$  : identifiant d'agents.

**Algorithme de rendez-vous**

Pour chaque quadruplet  $\varphi_k = (L_1, S_1, L_2, S_2)$  faire

Agent  $L_1$  suit la séquence de port  $S_1$

Si  $S_2$  correspond aux ports sortant de  $S_1$  alors

Agent  $L_1$  calcule la route de  $L_2$  jusqu'à  $\varphi_{k-1}$

Agent  $L_1$  rallonge sa route pour former un tunnel  
avec la route de  $L_2$

Sinon Agent  $L_1$  retourne à sa position de départ

L'Agent  $L_2$  opère de la même façon en inversant  $S_1$  et  $S_2$ .

# Quelques problèmes ouverts

## Problème ouvert 1

Est-ce-que le rendez-vous est possible dans le cas général en temps polynomial ?

## Problème ouvert 2

Est-ce-que le rendez-vous est toujours possible si la création des agents sur le réseau n'est pas simultanée ?

## Quelques perspectives

### Choix des numéro de ports

Est-ce-qu'un choix intelligent pour la numérotation des ports peut faciliter le rendez-vous ?

### Limitation de la capacité de calcul des agents

Est-ce-que le rendez-vous est toujours possible si les agents ont une mémoire finie (par exemple logarithmique en la taille du graphe dans le cas des graphes finis) ?