

Recherche décentralisée de connexité pour réseaux de capteurs mobiles

Merwan Achibet

Université du Havre

Jeudi 16 février 2012

Sommaire

- 1 Introduction
- 2 Déterminer la connexité
- 3 Créer la connexité

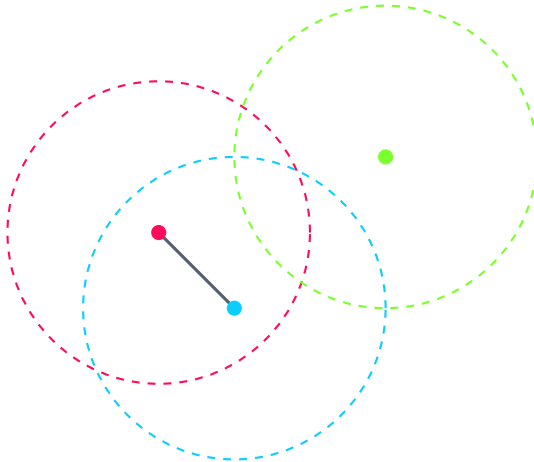
Problème

- Des capteurs sont lâchés dans un espace
- Doivent être proches pour communiquer
- Sont mobiles
- Sont vulnérables aux pannes
- → Un réseau dynamique

Deux questions :

- ① Comment savoir si le réseau est connexe ?
- ② Comment rendre le réseau connexe ?

Et ce, de manière décentralisée !



- ▶ ● et ● sont connectés
- ▶ ● est isolé

Sommaire

- 1 Introduction
- 2 Déterminer la connexité
- 3 Créer la connexité

Cadre

- On se place du point de vue d'un capteur c
 - On considère que le réseau est connexe si c pense que le réseau est connexe
-
- Peut être ramené à un problème de comptage décentralisé
 - Les capteurs connaissent n
 - Un capteur sait que \tilde{n} nœuds sont sur sa composante connexe
 - Si $\tilde{n} = n$, le réseau est connexe

Idée

- Chaque capteur surveille son voisinage et entretient sa propre vision du réseau
- Chaque information de présence est étiquetée temporellement
- Chaque capteur partage régulièrement sa vision du système
- À chaque partage, on évalue les avis puis soit on intègre, soit on ignore

Dans la mémoire interne de chaque capteur

N

Contient les voisins de c .

Exemple : $\{b, f, z\}$

Dans la mémoire interne de chaque capteur

C

Contient les nœuds que c pensent connectés au réseau. Ses entrées sont des triplets (x, t, s) avec

- x le nœud dont l'on suppose la présence
- t le temps de sa dernière détection
- s le nœud ayant transmis l'information (un voisin)

Exemple : $\{(a, 23, c), (f, 5, a)\}$

Dans la mémoire interne de chaque capteur

D

Contient les nœuds dont c doute de l'appartenance au réseau. Ses entrées sont des paires (x, t) avec

- x le nœud dont l'on suppose l'absence
- t le temps auquel son absence a été détectée

Exemple : $\{(p, 103)\}$

Modification par ses propres observations

c reçoit un message d'un nouveau voisin c'

$c' \rightarrow N, C$

Pas de message d'un voisin c' enregistré depuis un certain temps (surveillance)

$N, C \rightarrow D$

Modification par évaluation des rumeurs

À chaque réception d'un message, on l'évalue par rapport à nos connaissances pour juger quelles informations intégrer et quelles informations ignorer.

Critère d'évaluation

L'étiquette temporelle

Quelques exemples

$C' = \{\dots, (a, 42, e)\}$ **et aucune mention de a dans C ou D**

$\rightarrow C = \{\dots, (a, 42, c')\}$

$C' = \{\dots, (a, 42, e)\}$ **et $C = \{\dots, (a, 28, f)\}$**

$\rightarrow C = \{\dots, (a, 42, c')\}$

Quelques exemples

$C' = \{\dots, (a, 42, e)\}$ **et** $D = \{\dots, (a, 78, f)\}$

On ignore l'information

$D' = \{\dots, (a, 42, e)\}$ **et** $C = \{\dots, (a, 28, f)\}$

$C = \{\dots\}$

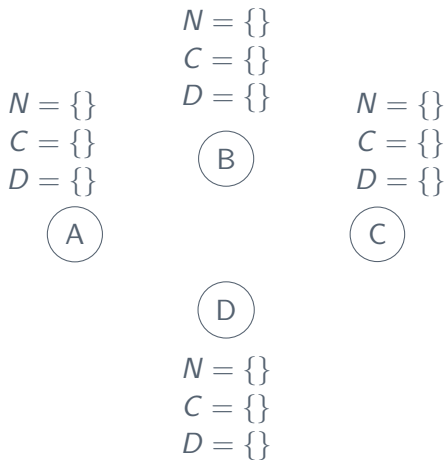
$D = \{\dots, (a, 42, c')\}$

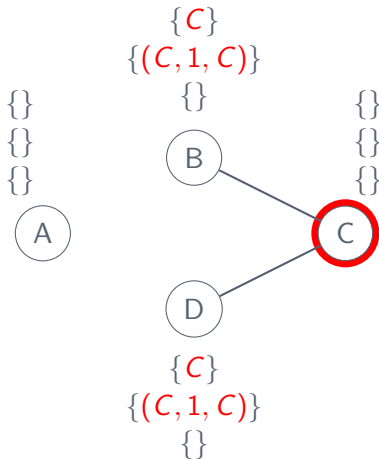
À quoi sert la source ?

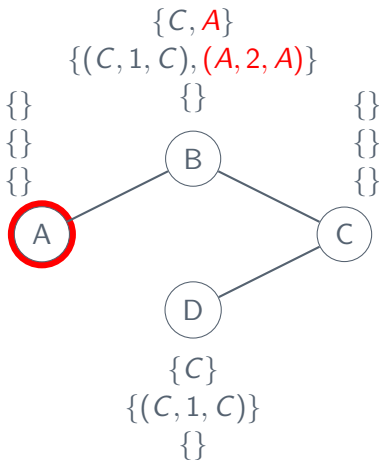
$D' = \{\dots, (b, 54, z)\}$ **et** $C = \{\dots, (a, 23, b), (b, 11, c)\}$

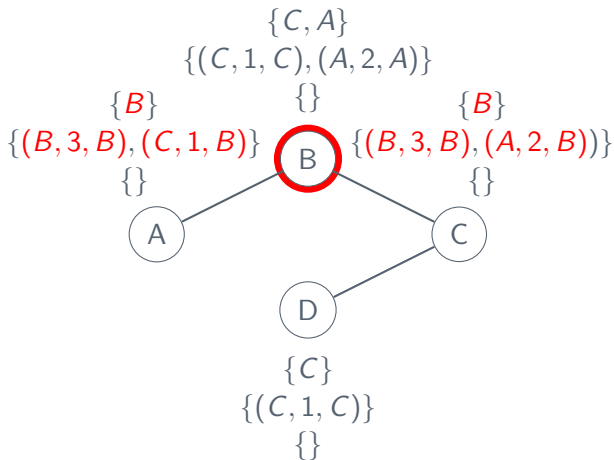
$(b, 11, c)$ passe de C à D

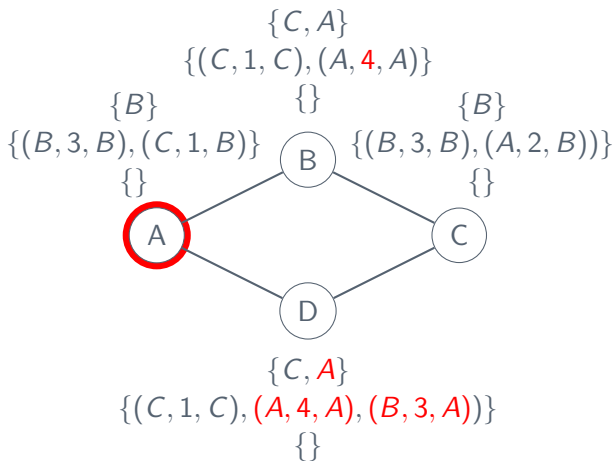
$(a, 23, b)$ aussi car b en est la source

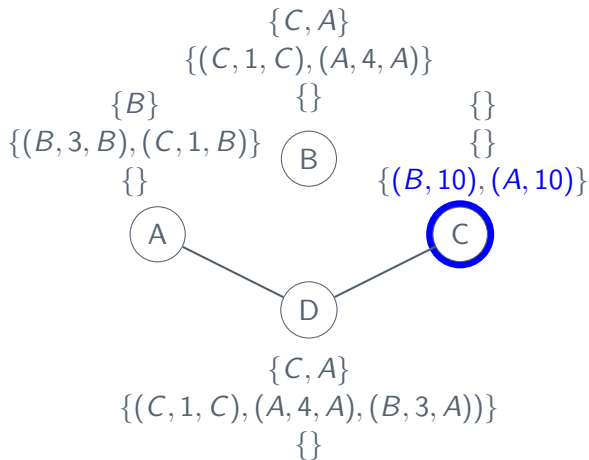


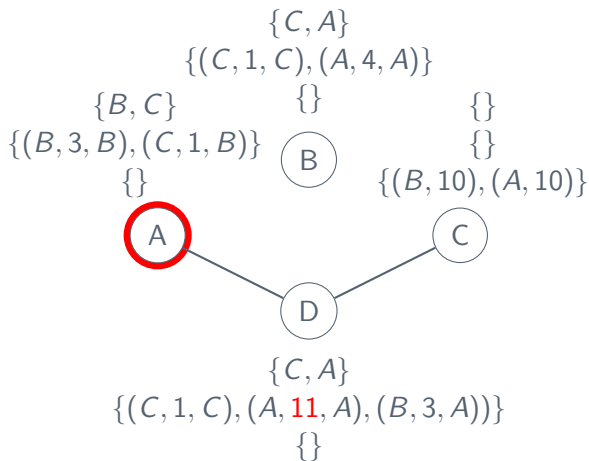


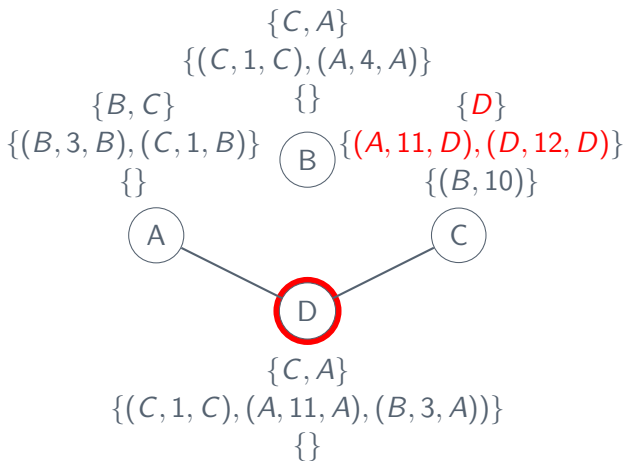












Sommaire

- 1 Introduction
- 2 Déterminer la connexité
- 3 Créer la connexité

Contraintes opposées

- Pour communiquer, ils doivent être proches
- Pour être efficaces, ils doivent être dispersés

→ On recherche un compromis équilibré.

Inspirations

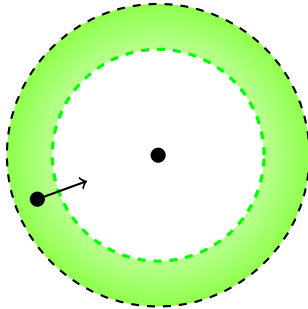
Boids

- Craig W. Reynolds, 1987
- Un jeu de règles simple
- Les actions locales...
- ... aboutissent à un comportement global

Systèmes particuliers

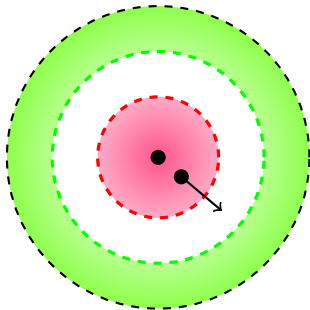
- Cheng, Cheng et Nagpal, 2005
- Forces de répulsion
- Répartition de particules dans des formes géométriques

L'attraction



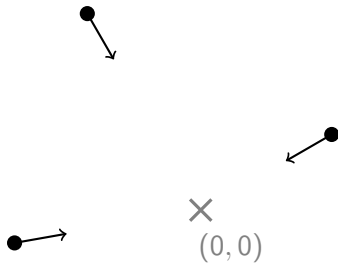
$$\vec{a} = \frac{\vec{p}_c - \vec{p}_v}{|\vec{p}_c - \vec{p}_v|^2}$$

La répulsion



$$\vec{r} = \frac{\vec{p} - \vec{p}_v}{|\vec{p} - \vec{p}_v|} (R_r - |\vec{p} - \vec{p}_v|)$$

La gravité



$$\vec{g} = -\frac{\vec{p}}{|\vec{p}|}$$

Combiner les différentes influences

$$\vec{f} = \frac{\vec{a} + \vec{r} + \vec{g}}{3} \quad (1)$$

- Chaque force a la même importance
- Au début de la simulation, acceptable
- Ensuite, le maillage s'affaisse

→ Démonstration

Combiner les différentes influences

Prioritiser les forces

Chaque capteur a une vitesse maximale.

- 1 On applique la 1ère force
- 2 S'il reste de la magnitude, on applique la 2nde force
- 3 S'il reste de la magnitude, on applique la 3ème force

Répulsion → Attraction → Gravité

→ Démonstration

Et les obstacles ?

- Une nouvelle force de répulsion capteur/obstacle
- En premier dans la liste de priorités

