

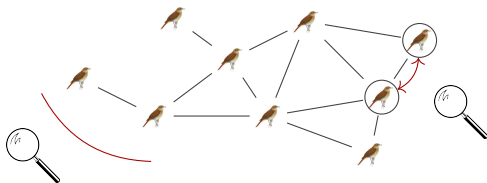
A brief introduction to **distributed algorithms**

Arnaud Casteigts
LaBRI, Université de Bordeaux

January 17, 2023

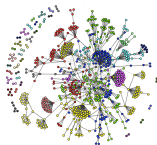
Séminaire du LOMA
(Laboratoire Ondes et Matière d'Aquitaine)

Theory of networks



Network as **data**

→ **centralized** algorithms...



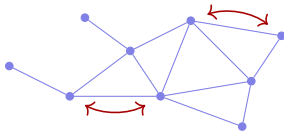
Network as **environment**

→ **decentralized** algorithms...
(a.k.a. **distributed**)



Distributed Algorithms

(Think globally, act locally)



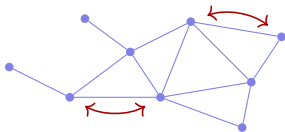
Collaboration of distinct entities to perform a common task.

No centralization available, interactions among neighbors.

Theoretical aspects of collective intelligence.

Distributed Algorithms

(Think globally, act locally)



Collaboration of distinct entities to perform a common task.

No centralization available, interactions among neighbors.

Theoretical aspects of collective intelligence.

Examples of problems:

Broadcast



Election



Spanning tree



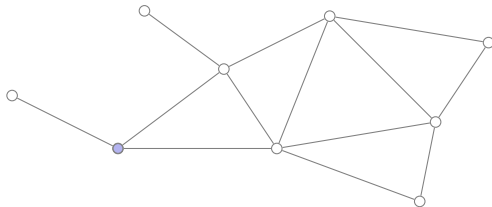
Counting



Consensus, naming, routing, exploration, coloring, dominating sets, ...

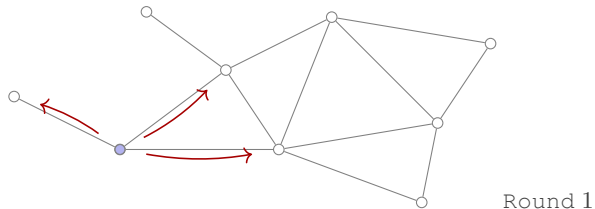
Example: Broadcasting \rightarrow Spanning tree \rightarrow Counting

Assumptions: synchronous communication / unique ids / distinguished node



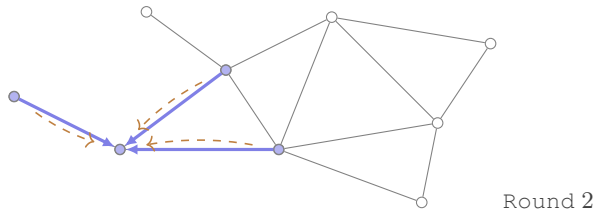
Example: Broadcasting \rightarrow Spanning tree \rightarrow Counting

Assumptions: synchronous communication / unique ids / distinguished node



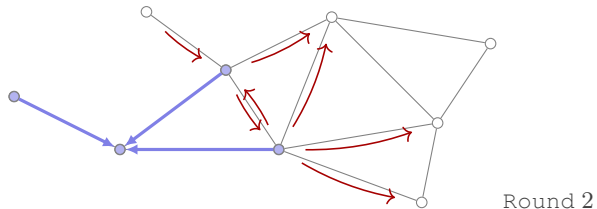
Example: Broadcasting \rightarrow Spanning tree \rightarrow Counting

Assumptions: synchronous communication / unique ids / distinguished node



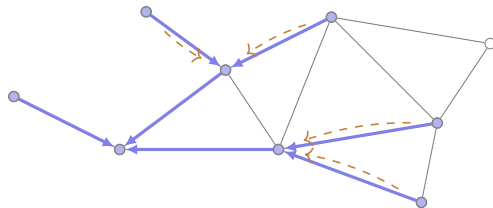
Example: Broadcasting \rightarrow Spanning tree \rightarrow Counting

Assumptions: synchronous communication / unique ids / distinguished node



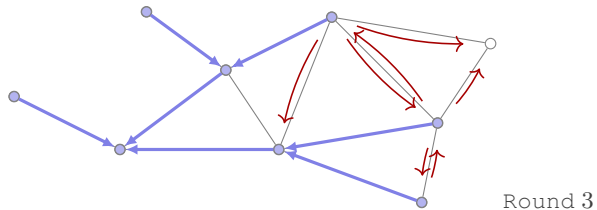
Example: Broadcasting \rightarrow Spanning tree \rightarrow Counting

Assumptions: synchronous communication / unique ids / distinguished node



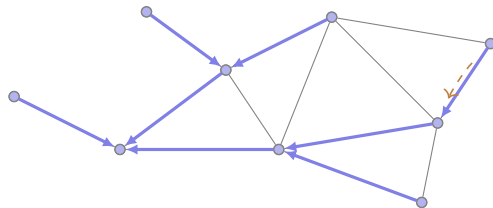
Example: Broadcasting \rightarrow Spanning tree \rightarrow Counting

Assumptions: synchronous communication / unique ids / distinguished node



Example: Broadcasting \rightarrow Spanning tree \rightarrow Counting

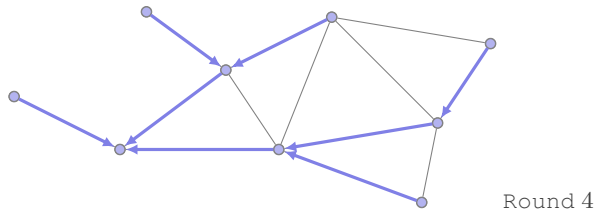
Assumptions: synchronous communication / unique ids / distinguished node



Round 4

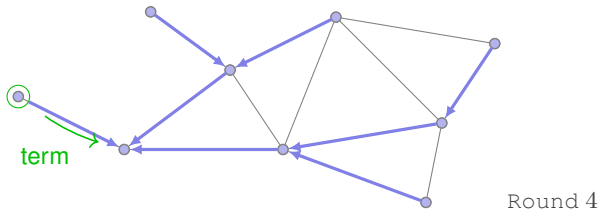
Example: Broadcasting \rightarrow Spanning tree \rightarrow Counting

Assumptions: synchronous communication / unique ids / distinguished node



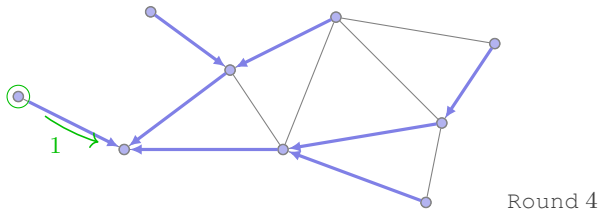
Example: Broadcasting \rightarrow Spanning tree \rightarrow Counting

Assumptions: synchronous communication / unique ids / distinguished node



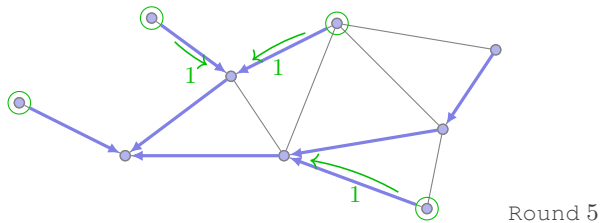
Example: Broadcasting \rightarrow Spanning tree \rightarrow Counting

Assumptions: synchronous communication / unique ids / distinguished node



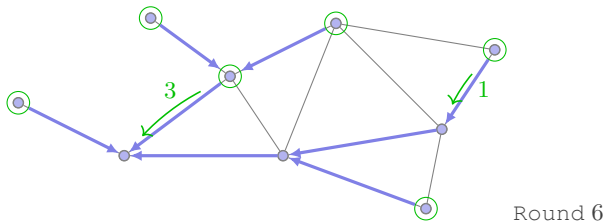
Example: Broadcasting \rightarrow Spanning tree \rightarrow Counting

Assumptions: synchronous communication / unique ids / distinguished node



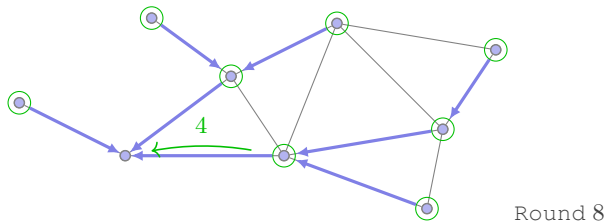
Example: Broadcasting \rightarrow Spanning tree \rightarrow Counting

Assumptions: synchronous communication / unique ids / distinguished node



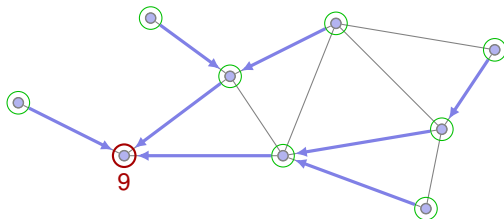
Example: Broadcasting \rightarrow Spanning tree \rightarrow Counting

Assumptions: synchronous communication / unique ids / distinguished node



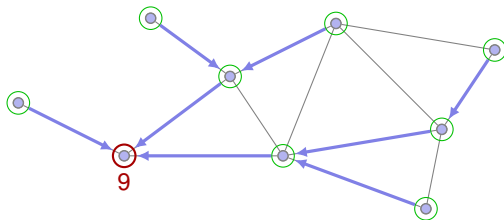
Example: Broadcasting \rightarrow Spanning tree \rightarrow Counting

Assumptions: synchronous communication / unique ids / distinguished node



Example: Broadcasting \rightarrow Spanning tree \rightarrow Counting

Assumptions: synchronous communication / unique ids / distinguished node



Time complexity: $O(\text{diameter}(G)) = O(n)$

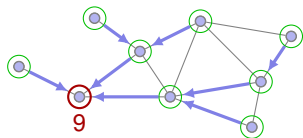
Message complexity: $O(m + n + n) = O(m)$

n : #nodes

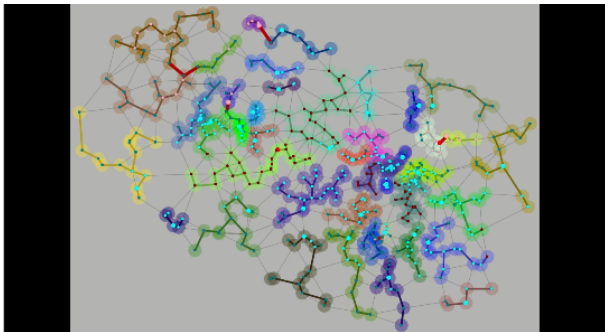
m : #edges

Example: Broadcasting \rightarrow Spanning tree \rightarrow Counting

Assumptions: synchronous communication / unique ids / distinguished node

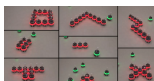
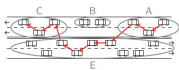


What if there are no distinguished nodes? (The GHS algorithm, 1983)



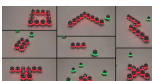
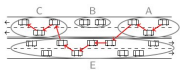
The world is distributed...

In technologies

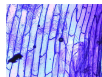
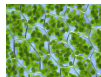


The world is distributed...

In technologies



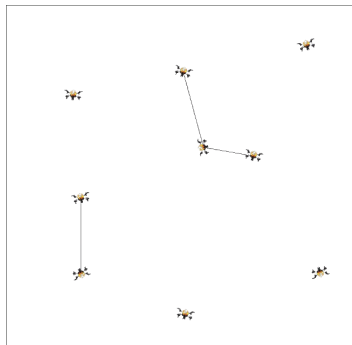
In nature



(Highly) dynamic networks?



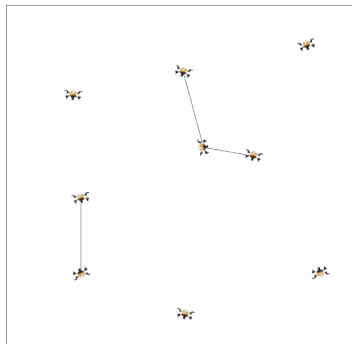
Example of scenario



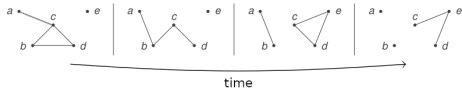
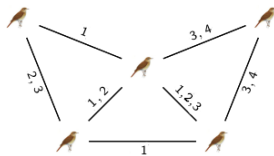
(Highly) dynamic networks?



Example of scenario



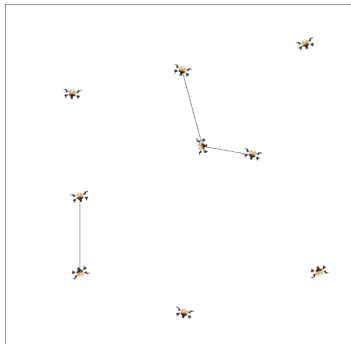
Modeling



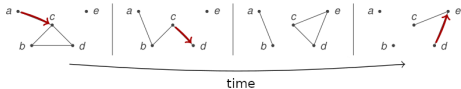
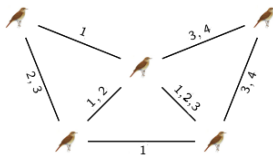
(Highly) dynamic networks?



Example of scenario



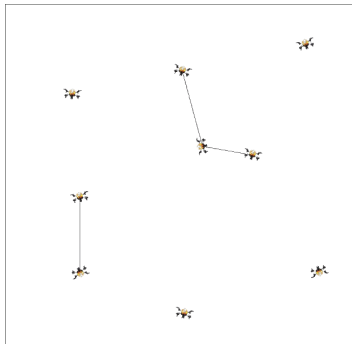
Modeling



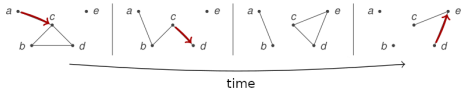
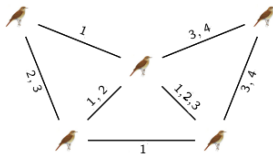
(Highly) dynamic networks?



Example of scenario



Modeling



Properties:

- ▶ Temporal connectivity?
- ▶ Repeatedly?
- ▶ Recurrent links?
- ▶ In bounded time?
- ▶ ...

TC

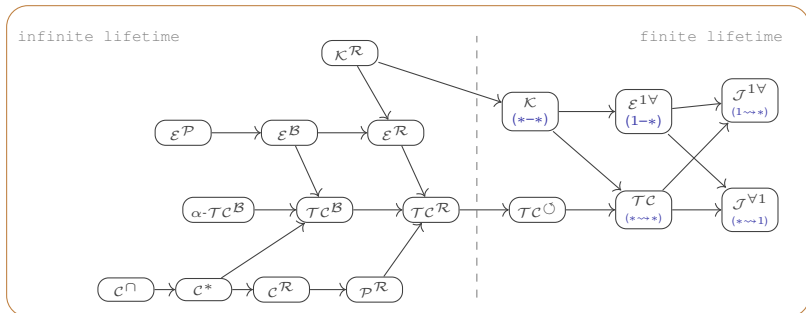
TC^R

\mathcal{E}^R

\mathcal{E}^B

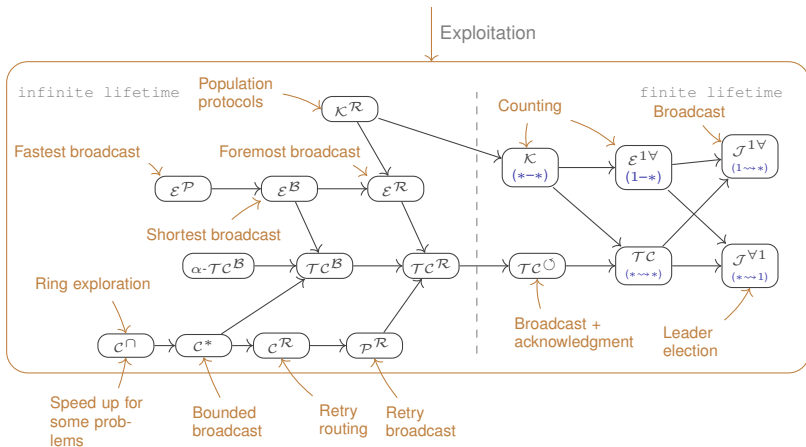
→ Classes of temporal graphs

Some classes of temporal graphs



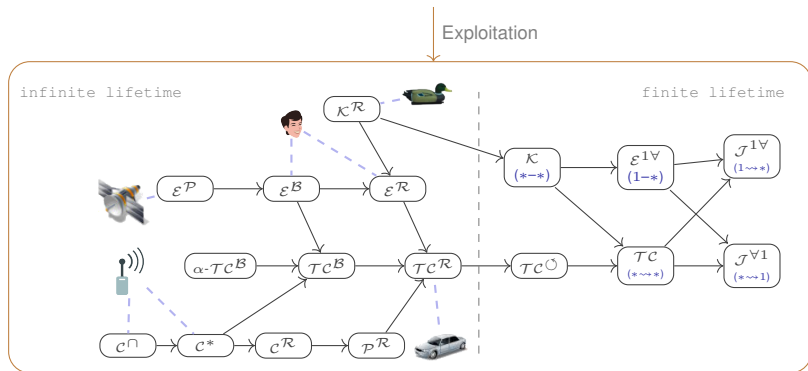
Some classes of temporal graphs

Distributed algorithm



Some classes of temporal graphs

Distributed algorithm

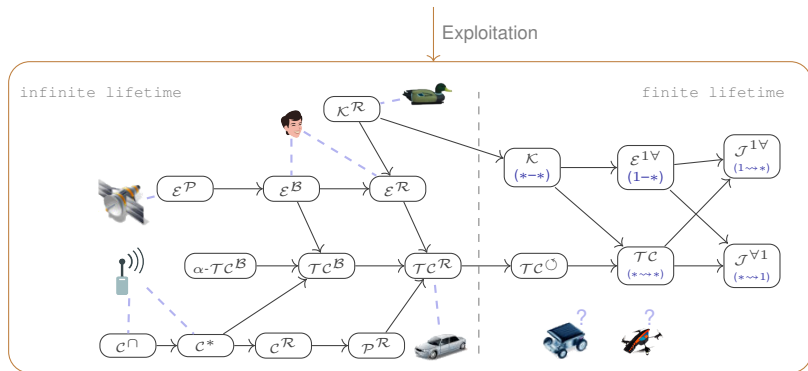


Analysis

Centralized algorithm

Some classes of temporal graphs

Distributed algorithm



Centralized algorithm

Movement synthesis

Theory fed by practical scenarios... and pedagogical practice!

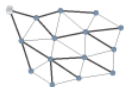
JBoTSIM: Collective intelligence / Network algorithms / Motion planning / Robotics



(h) Acceleration constraints



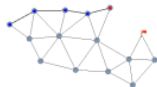
(k) Obstacle management



(a) Data aggregation



(b) Vehicular networks



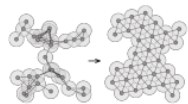
(c) Geographical routing



(o) Navigation (embedding)



(g) Territory sharing



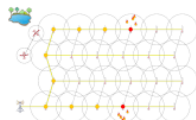
(d) Deployment by virtual forces



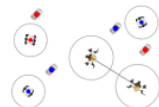
(e) Travelling Salesman Problem



(f) Toroidal space



(j) Fire-fighting aircrafts



(i) Heterogeneous park cleaning

Let's finish with the beginning... (first algorithm in literature)

Election in a **directed cycle**, with **synchronous communication** and **unique identifiers**
(Le Lann, Chang and Roberts, 1970's)

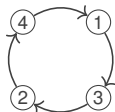
The node with the highest identifier gets elected.

onStart():

```
send(myID)
```

onMessage(otherID):

```
if (otherID > myID){  
    send(otherID)  
}else{  
    if (otherID == myID){  
        ELECTED ← true  
    }  
}
```



Let's finish with the beginning... (first algorithm in literature)

Election in a **directed cycle**, with **synchronous communication** and **unique identifiers**
(Le Lann, Chang and Roberts, 1970's)

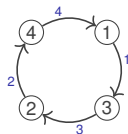
The node with the highest identifier gets elected.

onStart():

```
send(myID)
```

onMessage(otherID):

```
if (otherID > myID){  
    send(otherID)  
}else{  
    if (otherID == myID){  
        ELECTED ← true  
    }  
}
```



ronde 1

Let's finish with the beginning... (first algorithm in literature)

Election in a directed cycle, with **synchronous communication** and **unique identifiers**
(Le Lann, Chang and Roberts, 1970's)

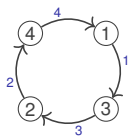
The node with the highest identifier gets elected.

onStart():

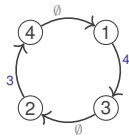
```
send(myID)
```

onMessage(otherID):

```
if (otherID > myID){
  send(otherID)
}else{
  if (otherID == myID){
    ELECTED ← true
  }
}
```



ronde 1



ronde 2

Let's finish with the beginning... (first algorithm in literature)

Election in a directed cycle, with **synchronous communication** and **unique identifiers**
(Le Lann, Chang and Roberts, 1970's)

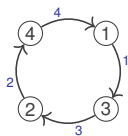
The node with the highest identifier gets elected.

onStart():

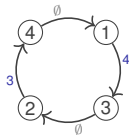
```
send(myID)
```

onMessage(otherID):

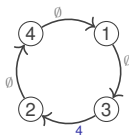
```
if (otherID > myID){
  send(otherID)
}else{
  if (otherID == myID){
    ELECTED ← true
  }
}
```



ronde 1



ronde 2



ronde 3

Let's finish with the beginning... (first algorithm in literature)

Election in a directed cycle, with **synchronous communication** and **unique identifiers**
(Le Lann, Chang and Roberts, 1970's)

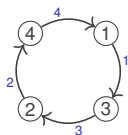
The node with the highest identifier gets elected.

onStart():

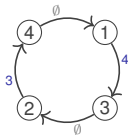
```
send(myID)
```

onMessage(otherID):

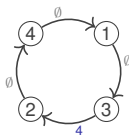
```
if (otherID > myID){  
  send(otherID)  
}else{  
  if (otherID == myID){  
    ELECTED ← true  
  }  
}
```



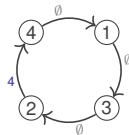
ronde 1



ronde 2



ronde 3



ronde 4

Let's finish with the beginning... (first algorithm in literature)

Election in a directed cycle, with **synchronous communication** and **unique identifiers**
(Le Lann, Chang and Roberts, 1970's)

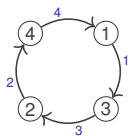
The node with the highest identifier gets elected.

onStart():

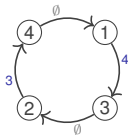
```
send(myID)
```

onMessage(otherID):

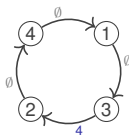
```
if (otherID > myID){  
    send(otherID)  
}else{  
    if (otherID == myID){  
        ELECTED ← true  
    }  
}
```



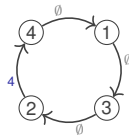
ronde 1



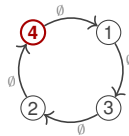
ronde 2



ronde 3



ronde 4



ronde 5

Let's finish with the beginning... (first algorithm in literature)

Election in a directed cycle, with **synchronous communication** and **unique identifiers**
(Le Lann, Chang and Roberts, 1970's)

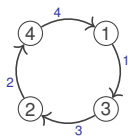
The node with the highest identifier gets elected.

onStart():

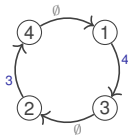
```
send(myID)
```

onMessage(otherID):

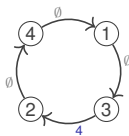
```
if (otherID > myID){
  send(otherID)
}else{
  if (otherID == myID){
    ELECTED ← true
  }
}
```



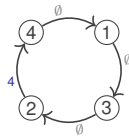
ronde 1



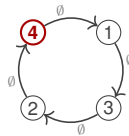
ronde 2



ronde 3



ronde 4



ronde 5

Time complexity? Message complexity ?



Thanks!