

# Back to the Coordinated Attack Problem

*The Power of Topology Tools*

E. Godard E. Perdereau

LIS – AMU

*Journées ANR Descartes 09/10*



LABORATOIRE  
D'INFORMATIQUE  
& SYSTÈMES

# Back to the Coordinated Attack Problem

## A Kind of Folklore Problem (Gray 78)

Two generals have gathered forces on top of two facing hills. In between, in the valley, their common enemy is entrenched. This enemy can only be defeated if both armies attack.

Every day each general sends a messenger to the other through the valley. However this is risky as the enemy may capture them. Now they need to get the last piece of information: are they both ready to attack?

It is also stated as the *Two Generals Problem*.

# More Modern (and Formal)

This is the **Consensus Problem** for **two** processes that communicate by **synchronous message-passing** with possible **omission faults**.

## The Consensus Problem

Each process is given an initial value. Each process has to decide a value such that

**Agreement** At most 1 value can be decided,

**Validity** The decided value is included in the set of initial values,

**Termination** Any *non-faulty* process eventually decides.

# A Key Impossibility Result

*one of the first distributed computability result*

J. Gray prove it was a problem impossible to solve deterministically.  
The proof is basically that, if the two generals want to communicate,  
they will be in an infinite sequence of mutual acknowledgements...

# An Interesting Case ?

- Consensus is a well known problem
- Simple (minimal!) instance
- Faults, but benign ones

Maybe too simple and obvious?...

# An Interesting Case ?

- Consensus is a well known problem
- Simple (minimal!) instance
- Faults, but benign ones

Maybe too simple and obvious?... Actually, yes!!

Because it is possible to lose **every message**, it is obvious the task is impossible.

# What is the Problem ?

In [Gray78], such a dramatic scenario is not explicitly ruled out...

In textbooks:

- Santoro's *Design and Analysis of Distributed Algorithms*: the zero-message scenario is explicitly excluded.
- Lynch's *Distributed Algorithms*: the problem statement is weakened such that it is not obviously impossible.

## Interesting Question

What are the impossible cases ?

# Describing Scenarios : Message Adversaries

Let  $\Sigma = \{\circ \leftrightarrow \bullet, \circ \leftarrow \bullet, \circ \rightarrow \bullet, \circ - \bullet\}$ ,

and  $\Gamma = \{\circ \leftrightarrow \bullet, \circ \leftarrow \bullet, \circ \rightarrow \bullet\}$ .

The interpretation is that :

- $\circ \leftrightarrow \bullet$ , no process loses messages
- $\circ \leftarrow \bullet$ , the message of process  $\circ$  is not transmitted
- $\circ \rightarrow \bullet$ , the message of process  $\bullet$  is not transmitted
- $\circ - \bullet$ , both messages are not transmitted.

## definition

A *message adversary* is a set of infinite sequences of elements of  $\Sigma$ .



# Examples

With the infinite word notation

- at each round, up to 2 messages can be lost:  $\Sigma^\omega$ .
- at each round, only one message can be lost:  $\Gamma^\omega$ .
- at most one of the processes can lose messages:  $S_1 = \{\circ\leftrightarrow\bullet, \circ\leftarrow\bullet\}^\omega \cup \{\circ\leftrightarrow\bullet, \circ\rightarrow\bullet\}^\omega$ .
- at most one of the processes can crash:  $G_1 = \{\circ\leftrightarrow\bullet\}^\omega \cup \{\circ\leftrightarrow\bullet\}^* (\{\circ\leftarrow\bullet\}^\omega + \{\circ\rightarrow\bullet\}^\omega)$ .
- The communication system is *fair*:  $Fair = \Sigma^\omega \setminus \Sigma^* (\{\circ\rightarrow\bullet, \circ\leftarrow\bullet\}^\omega \cup \{\circ\rightarrow\bullet, \circ\rightarrow\bullet\}^\omega)$ .

# Distributed Computability

Finally and classically,

## Definition

A algorithm  $\mathcal{A}$  solves the Coordinated Attacked Problem for the message adversary  $L$  if for any condition  $w \in L$ , there exists  $u \in \text{Pref}(w)$  such that the states of the two processes ( $s^\circ(u)$  and  $s^\bullet(u)$ ) satisfy the three conditions of Consensus.

## Definition

A message adversary  $L$  is said to be *solvable* if there exists an algorithm that solves the Coordinated Attacked Problem for  $L$ . It is said to be an *obstruction* otherwise.

# Obstructions

## Restating the Initial Problem

What are the obstructions for the Coordinated Attack Problem?  
Which are minimal ?

# Summary

## Minimal obstruction

- $\Gamma^\omega$  is an obstruction set.
- (almost) all subset of  $\Gamma^\omega$  are not solvable.
  
- 2011 : purely combinatorial proof
- this talk : topological proof

# Index of a Scenario




- $\delta(o \rightarrow \bullet) = -1$ ,
- $\delta(o \leftrightarrow \bullet) = 0$ ,
- $\delta(o \leftarrow \bullet) = 1$ .

## Definition

Let  $w \in \Gamma^*$ . We define  $ind(\varepsilon) = 0$ . If  $|w| \geq 1$ , then we have  $w = ua$  where  $u \in \Gamma^*$  and  $a \in \Gamma$ . In this case, we define

$$ind(w) := 3ind(u) + (-1)^{ind(u)}\delta(a) + 1.$$

# Examples

word of length "1"			
index	0	1	2

# Examples

word of length "1"	○→●	○↔●	○←●
index	0	1	2

word of length "2"	○→●○→●	○→●○↔●	○→●○←●
index	0	1	2

word of length "2"	○↔●○→●	○↔●○↔●	○↔●○←●
index	5	4	3

word of length "2"	○←●○→●	○←●○↔●	○←●○←●
index	6	7	8

# Distributed Computability

## Theorem (Fevat, G. 2011)

Let  $L \subset \Gamma^\omega$ , then Consensus is solvable for message adversary  $L$  if and only if one of the following holds

- $\exists f \in \text{Fair}, f \notin L,$
- $\exists (u, u') \in \text{SPair}, u, u' \notin L,$
- $\circ \rightarrow \bullet^\omega \notin L,$
- $\circ \leftarrow \bullet^\omega \notin L.$

where we define *special pairs* to be

$$\text{SPair} = \{(w, w') \in \Gamma^\omega \times \Gamma^\omega \mid w \neq w', \forall r \in \mathbb{N} | \text{ind}(w|_r) - \text{ind}(w'|_r) | \leq 1\}.$$



# Classical Proof Technique

## Necessary Condition

It is a classical proof by **bivalency**.

## Sufficient Condition

A new index-based Consensus Algorithm.

# A Consensus Algorithm $\mathcal{A}_w$

```

r=0; initother=null;
if " $\odot = \circ$ " then
  | ind=0;
else
  | ind=1;
while " $|ind - ind(w|_r)| \leq 1$ " do
  | msg = (init,ind);
  | send(msg); msg = receive();
  | if msg == null then // message was lost
  | | " $ind = 3 * ind$ ";
  | else
  | | " $ind = 2 * msg.ind + ind$ ";
  | | initother = msg.init;
  | r=r+1;

```

# A Consensus Algorithm $\mathcal{A}_w$ (cont.)

```

if " $\odot = \circ$ " then
  | if " $ind \leq ind(w|_r)$ " then
  | | Output: init
  | else
  | | Output: initother
else
  | if " $ind \geq ind(w|_r)$ " then
  | | Output: init
  | else
  | | Output: initother

```

# Fundamental Invariant

## Proposition

For any round  $r$  of an execution of Algorithm  $\mathcal{A}_w$  under scenario  $v \in \Gamma^r$ , such that no process has already halted,

$$\begin{cases} |ind_r^\bullet - ind_r^\circ| = 1, \\ sign(ind_r^\bullet - ind_r^\circ) = (-1)^{ind(v)}, \\ ind(v) = \min\{ind_r^\circ, ind_r^\bullet\}. \end{cases}$$

# Simplicial Complexes

## The real ones

### Definition

$\sigma = \{v_0, \dots, v_n\} \subset \mathbb{R}^N$  is a *simplex* of dimension  $n$  if the vector space generated by  $\{v_1 - v_0, \dots, v_n - v_0\}$  is of dimension  $n$ .

### Definition

A *simplicial complex* is a collection  $\mathcal{C}$  of *simplices*

- If  $\sigma \in \mathcal{C}$  and  $\sigma' \subseteq \sigma$ , then  $\sigma' \in \mathcal{C}$ ,
- If  $\sigma, \tau \in \mathcal{C}$  and  $|\sigma \cap \tau| \neq \emptyset$  then there exists  $\sigma' \in \mathcal{C}$  such that  $|\sigma \cap \tau| = |\sigma'|$ .

*Abstract simplicial complexes* are an equivalent presentation when the collection is finite.

# Geometric Realization

$|\sigma|$  is the convex hull of  $\sigma$ .

It is the *geometric realization* of  $\sigma$ .

# Chromatic Simplex

Given a set  $P$ , a chromatic simplex is  $(\sigma, c)$  where  $c : \sigma \rightarrow P$  is injective

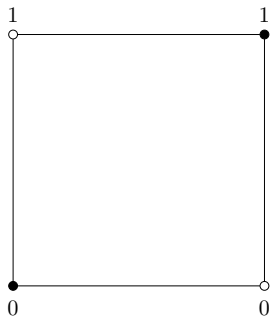
# Encoding the Global State of a Distributed System

$P$  is the set of processes (here  $P = (\circ, \bullet)$ ),  $state^\circ$  is the state of  $\circ$ .

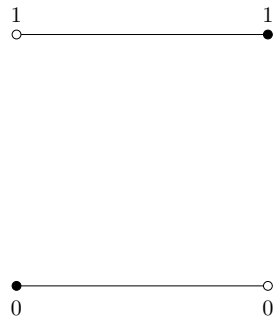
- $(\circ, state^\circ) \Rightarrow$  **local state** of  $\circ$
- the simplex  $\{(\circ, state^\circ), (\bullet, state^\bullet)\} \Rightarrow$  global state



# Encoding the Binary Consensus Problem

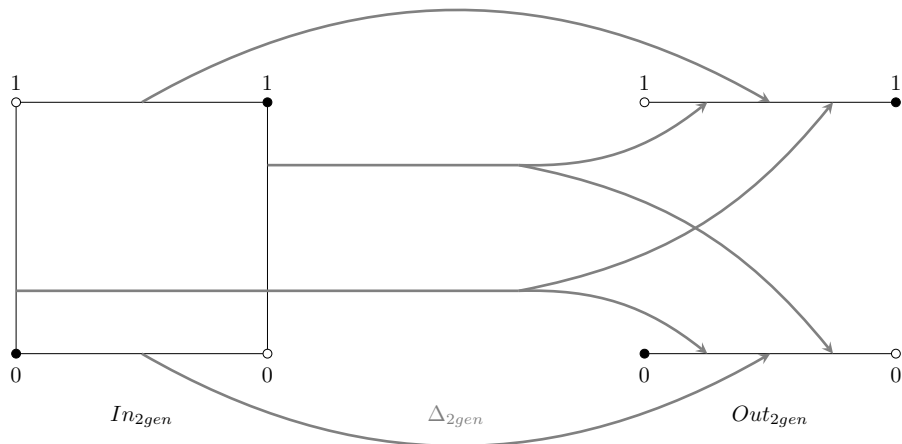


$In_{2gen}$



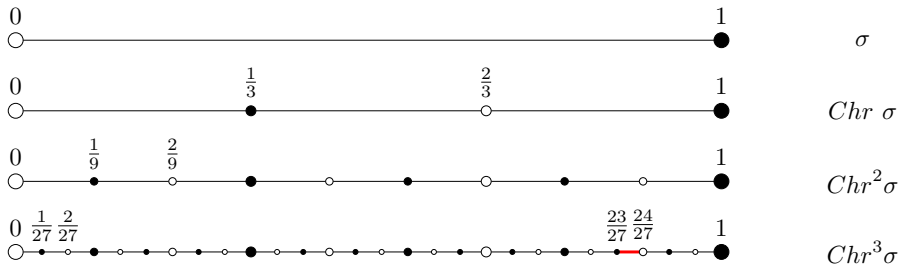
$Out_{2gen}$

# Encoding the Binary Consensus Problem



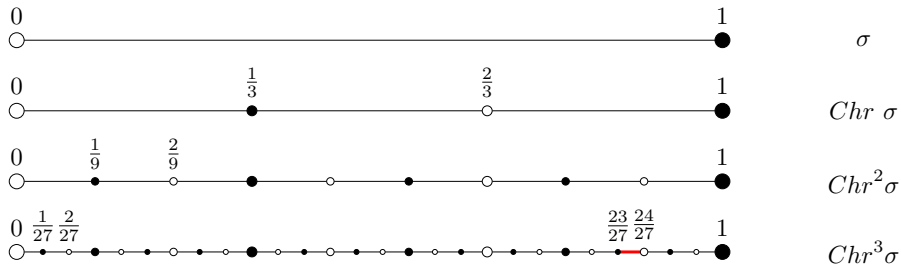
# Chromatic Subdivision

It is a complex operator *Chr* such that



# Chromatic Subdivision

It is a complex operator  $Chr$  such that

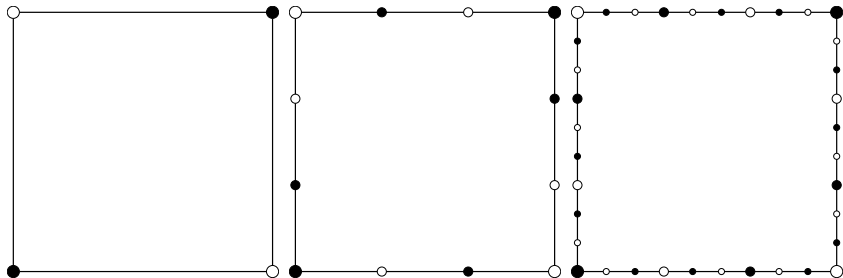


## Encoding Partial Executions

Let  $w \in \Gamma^*$ , the simplex  $\left\{ \frac{ind(w)}{3||w||}, \frac{ind(w)+1}{3||w||} \right\}$  is associated to  $w$ .

$\left[ \frac{23}{27}, \frac{8}{9} \right] \Rightarrow$  corresponds to  $\circ \leftarrow \bullet \circ \leftrightarrow \bullet \circ \rightarrow \bullet$ .

# Iterated Protocol Complex



$PC^L(r)$  is the protocol complex for  $L$  at round  $r$ .

**Uncertainty** appears as **adjacent** simplices

# Encoding Full Executions

- $\forall n \in \mathbb{N}, \forall w \in \Gamma^n \quad ind_n(w) = \frac{ind(w)}{3^n}$
- $\forall w \in \Gamma^\omega \quad \overline{ind}(w) = \lim_{n \rightarrow +\infty} ind_n(w|_n)$

# Encoding Full Executions

- $\forall n \in \mathbb{N}, \forall w \in \Gamma^n \quad ind_n(w) = \frac{ind(w)}{3^n}$
- $\forall w \in \Gamma^\omega \quad \overline{ind}(w) = \lim_{n \rightarrow +\infty} ind_n(w|_n)$

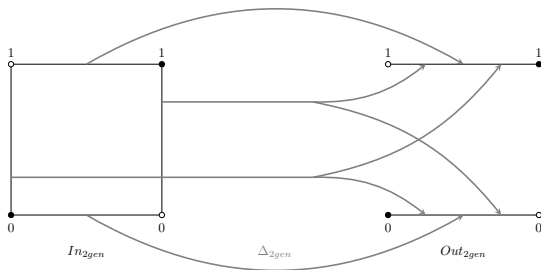
With  $L \subset \Gamma^\omega$ , we associate a subspace of  $[0, 1]$  to  $L$ .

## Special Pairs

$\{w, w'\}$  is a special pair iff  $\overline{ind}(w) = \overline{ind}(w')$

# First Impossibility Proof

For  $\Gamma^\omega$ , remark that the image of  $\Gamma$  by  $\overline{ind}$  is  $[0, 1]$  :



If an algorithm exists, then we have a morphism from a connected space onto a disconnected one.



# Terminating Subdivision

(inspired by Gafni, Kuznetsov Manolescu 2014)

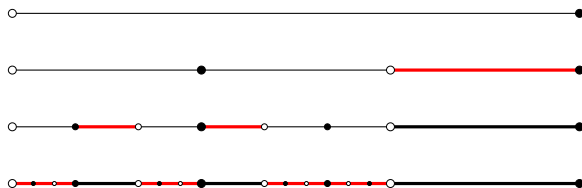
## Definition

Let  $C$  be a chromatic complex. A *terminating subdivision*  $TS$  of  $C$  is a (possibly infinite) sequence of chromatic complexes  $(\Sigma_k)_{k \in \mathbb{N}}$  such that

- $\Sigma_0 = \emptyset$ ,
- for all  $k \geq 1$   $\Sigma_k \subset \text{Chr}^k C$ ,
- $\cup_{i \leq k} \Sigma_i$  is a simplicial complex.

$K(TS) = \cup_{i \leq 0} \Sigma_i$  is the terminating complex (could be infinite).

# Terminating Subdivision



$$\Sigma_0 = \emptyset$$

$$\Sigma_1 \equiv \{o \leftarrow \bullet\}$$

$$\Sigma_2 \equiv \{o \leftrightarrow \bullet \leftarrow \bullet, o \rightarrow \bullet \leftrightarrow \bullet\}$$

$\Sigma_3$  is all remaining simplices

# Admissible Terminating Subdivisions

## Definition

A terminating subdivision  $TS$  is *admissible* for  $L$  if for any scenario  $\rho \in L$  the corresponding sequence of edges  $\sigma_0, \sigma_1, \dots$  is such that there exists  $r > 0$  and a simplex  $\tau \in K(TS)$  such that  $|\sigma_r| \subseteq |\tau|$ .

# Result

## Theorem (G., Perdereau 2019)

Consensus is solvable in  $L \subseteq \Gamma^\omega$  if and only if there exist a terminating subdivision  $\Phi$  of  $In_{2gen}$  and a simplicial function  $\delta: K(\Phi) \rightarrow \{0, 1\}$  such that :

- $\Phi$  is admissible for  $L$ ;
- For all simplex  $\sigma \in In_{2gen}$ , if  $\tau \in K(\Phi)$  is such that  $|\tau| \subset |\sigma|$ , then  $\delta(\tau) \in \Delta'_{2gen}(\sigma)$ ;
- $|\delta|$  is continuous.

# Necessary Condition

Suppose we have an algorithm  $\mathcal{A}$  for  $L$ .

$\Sigma_r = \{\{x, y\} \in PC^L(r) \mid x \text{ and } y \text{ have both decided and at least one has just decided in round } r\}$

$\delta(x) = \textit{decision}(x) \quad \forall x \in V(\Sigma_r), \textit{ piecewise}$

# Necessary Condition

Suppose we have an algorithm  $\mathcal{A}$  for  $L$ .

$\Sigma_r = \{ \{x, y\} \in PC^L(r) \mid x \text{ and } y \text{ have both decided and at least one has just decided in round } r \}$

$\delta(x) = \text{decision}(x) \quad \forall x \in V(\Sigma_r), \text{ piecewise}$

- $\Phi = (\Sigma_r)$  is admissible for  $L$
- to prove  $|\delta|$  continuous, we show that  $\forall x \in |K(\Phi)|$

$$\exists \eta_x > 0 \quad \forall y \in |K(\Phi)| \quad |x - y| \leq \eta_x \Rightarrow |\delta|(x) = |\delta|(y)$$

with  $\eta_x = \min \{ \frac{1}{3^{r+1}} \mid \exists r \in \mathbb{N}, \exists y, \{x, y\} \in V(\Sigma_r) \}$

# Algorithm

Since  $|\delta|$  is continuous, there exists  $\eta(x)$  such that for all  $y$ ,  
 $\|x - y\| < \eta(x) \Rightarrow \delta(x) = \delta(y)$ .

**Data:** function  $\eta$

**Input:**  $init \in \{0, 1\}$

$t = 1$ ;

**if**  $\odot = \bullet$  **then**

$ind = 1$ ;

$initw = \text{null}$ ;

$initb = init$ ;

**else**

$ind = 0$ ;

$initw = init$ ;

$initb = \text{null}$ ;

# Algorithm

**while**

$geo(ind/t, initw, initb) \notin |K(\Phi)| \vee \eta(geo(ind/t, initw, initb)) < t$

**do**

msg = (init,ind); send(msg); msg = receive();

**if**  $msg == null$  **then** // message was lost

$ind = 3 * ind$ ;

**else**

$ind = 2 * msg.ind + ind$ ;

**if**  $\odot = \bullet$  **then**

$initw = msg.init$ ;

**else**

$initb = msg.init$ ;

$t = t/3$ ;

**Output:**  $|\delta|(ind/t)$



# Interpretation

The combinatorial description is better explained by connectivity

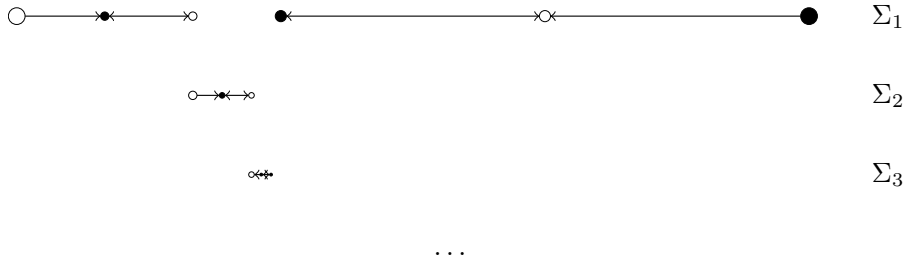
## Combinatorial Theorem

Let  $L \subset \Gamma^\omega$ , then Consensus is solvable for message adversary  $L$  if and only if one of the following holds

- $\exists f \in \text{Fair}, f \notin L,$
- $\exists (u, u') \in \text{SPair}, u, u' \notin L,$
- $\circ \rightarrow \bullet^\omega \notin L,$
- $\circ \leftarrow \bullet^\omega \notin L.$

# Why special pairs are tricky

Consider  $L = \Gamma^\omega \setminus \{ \circ \rightarrow \bullet \circ \leftarrow \bullet^\omega \}$



# Conclusion

- Consensus is not solvable for  $\Gamma^\omega$  : bivalency  $\Leftrightarrow$  connectivity
- Computability for arbitrary message adversaries
- topology tools are powerful, but **beware of definition of simplicial complexes**
- Extension to  $n > 2$  : presented at PODC 2019 by Nowak, Schmid, and Winkler

# Questions ?

Thanks for your attention.