Counting with Population Protocols

Yves Mocquard\textsuperscript{1}, Emmanuelle Anceaume\textsuperscript{2}, James Aspnes\textsuperscript{3}, Yann Busnel\textsuperscript{4} and Bruno Sericola\textsuperscript{5}

\textsuperscript{1}Université de Rennes 1, France
\textsuperscript{2}CNRS / IRISA, France
\textsuperscript{3}Yale University, USA
\textsuperscript{4}Ensai, France
\textsuperscript{5}Inria, France
Population protocol model

Introduced by Angluin, Aspnes, Diamadi, Fischer and Peralta in 2004

A model of distributed systems with minimal set of assumptions

1. Finite-state agents / automata
2. Agents have no identities
3. Communication is occasionally possible between agents

→ **Uniformity** All the agents execute the same code and this code is independent of the population size
→ **Anonymity**: No room for agents to store a unique identifier

---

A motivating example: birds [Angluin04]

A flock of birds on which are strapped identical sensors
When birds are close together, their sensors can interact and compute simple functions

- Decide if at least five birds have elevated body temperature
- Decide if at least five percent of the birds have elevated body temperature
- Compute the proportion of birds that have elevated temperature
- ...

NCA 2015, Boston, MA, USA
Motivation to study the population protocol model
Model
The counting problem
Performance evaluation
Conclusion and future works
The population protocol model (informal)

- Collection of finitely many finite-state agents
- Agents are indistinguishable (identical program + no identity)
- Agents enter their initial state by applying an input function $I$ to a finite set $X$ of input values

\[
\begin{align*}
I(\text{sick}) &= q_1 \\
I(\text{safe}) &= q_0
\end{align*}
\]
The population protocol model (informal)

- Collection of **finitely many** finite-state agents
- Agents are **indistinguishable** (identical program + no identity)
- Agents enter their **initial state** by applying an input function $I$ to a **finite set** $X$ of input values
- When two agents interact, they apply an **interaction function** $f$
The population protocol model (informal)

- Collection of **finitely many** finite-state agents
- Agents are **indistinguishable** (identical program + no identity)
- Agents enter their initial state by applying an input function $I$ to a finite set $X$ of input values
- When two agents interact, they apply an interaction function $f$
The population protocol model (informal)

- Collection of **finitely many** finite-state agents
- Agents are **indistinguishable** (identical program + no identity)
- Agents enter their **initial state** by applying an input function \( I \) to a **finite set** \( X \) of input values
- When two agents interact, they apply an **interaction function** \( f \)
- Output value = Decoding function \( O \) of agent current state

Are there more than 5 sick birds?
The population protocol model (informal)

- Collection of **finitely many** finite-state agents
- Agents are **indistinguishable** (identical program + no identity)
- Agents enter their **initial state** by applying an input function \( I \) to a **finite set** \( X \) of input values
- When two agents interact, they apply an **interaction function** \( f \)
- Output value = Decoding function \( O \) of agent current state
The population protocol model (informal)

- Collection of **finitely many** finite-state agents
- Agents are **indistinguishable** (identical program + no identity)
- Agents enter their **initial state** by applying an input function \( I \) to a finite set \( X \) of input values
- When two agents interact, they apply an interaction function \( f \)
- Output value = Decoding function \( O \) of agent current state

All agents eventually converge to a correct common or distributed output value
The population protocol model (informal)

- Collection of finitely many finite-state agents
- Agents are indistinguishable (identical program + no identity)
- Agents enter their initial state by applying an input function $I$ to a finite set $X$ of input values
- When two agents interact, they apply an interaction function $f$
- Output value = Decoding function $O$ of agent current state

All agents eventually converge to a correct common or distributed output value
A population configuration $C$ specifies the state of each agent.

**Fairness condition**: enforces that any reachable configuration is eventually reached.

A stable computation is an infinite fair sequence of configurations that **converges to a correct common or distributed output value**.

- Convergence is a global property: agents generally do not know that convergence has been reached.
- With suitable stochastic assumptions, it is possible to determine the number of interactions until the output stabilizes.
A flock of birds on which are strapped identical sensors
When birds are close together, their sensors can interact and compute simple functions

Decide if at least five birds have elevated body temperature?
Decide if at least five percent of the birds have elevated body temperature?
Compute the proportion of birds that have elevated temperature?
A predicate can be seen as a function that returns true or false.
Predicates can be written as $P(x_1, x_2, \ldots, x_k)$, where
- $k = \text{number of possible initial states}$
- $x_i = \text{number of agents starting in the } i\text{-th state}$

Examples:
- the "count-to-5" bird protocol: $P(x_1, x_2) = \text{true iff } x_1 \geq 5$
- the "majority" bird protocol: $P(x_1, x_2) = \text{true iff } x_1 \geq x_2$
- the "parity" bird protocol: $P(x_1, x_2) = \text{true iff } x_1 \equiv 0(\text{mod}2)$
Computational power of population protocols

Theorem (Angluin et al. 2007)

A predicate is computable by a population protocol if and only if it is in one of the following forms\(^a\):

- \(\sum_{i=1}^{k} c_i x_i \geq a\)
- \(\sum_{i=1}^{k} c_i x_i \equiv a \pmod{b}\),

where \(a_i, b\) and \(c_i's\) are integer constant

And every boolean combination of these predicates

Our contribution: counting the exact percentage of sick and safe birds

Problem

Design of a population protocol that exactly determines the difference between the percentage of sick birds and the percentage of safe birds.
Our algorithm to compute the exact difference between the percentage of \#red and \#blue

Algorithm

- **Finite input alphabet** $X = \{\text{sick, safe}\}$
- **Input function** $I$: $I(\text{sick}) = m$, $I(\text{safe}) = -m$
- **Finite set of states** $Q = \{-m, -m + 1, \ldots, m - 1, m\}$
- **Output function** $O$: $O(q) = \lfloor 100q/m + 1/2 \rfloor$
- **Transition function**
  
  $f : (q_1, q_2) \rightarrow (\lfloor (q_1 + q_2)/2 \rfloor, \lceil (q_1 + q_2)/2 \rceil)$
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?

\[ \sum_{i=1}^{5} = \text{constant} = -20. \]

As you will see this is an invariant of the protocol.
What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
Problem
What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
Problem
What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?
Problem

What is the difference between the percentage of sick and the percentage of safe birds?

\[
\left\lfloor \frac{100q}{m} + \frac{1}{2} \right\rfloor = \left\lfloor \frac{100 \times (-4)}{20} + \frac{1}{2} \right\rfloor = -20
\]
The choice of the interacting agents is a **non-deterministic choice**.

**Probabilistic assumptions:** interactions are orchestrated by a uniform fair random scheduler:

- At each discrete time \( t \), any two agents \( i \) and \( j \) are randomly chosen from a uniform distribution \( p_{i,j}(t) \), with
  
  \[
  p_{i,j}(t) = \frac{1}{n(n-1)}
  \]
Main ideas of the convergence proof.

- We show that at each step, the sum of the agents’ state is constant. For every \( t \geq 0 \),

\[
\sum_{i=1}^{n} C_t^{(i)} = \sum_{i=1}^{n} C_0^{(i)}
\]

- Let \( \ell = \frac{1}{n} \sum_{i=1}^{n} C_t^{(i)} \) and \( L = (\ell, \ldots, \ell) \), we show that

\[
E \left( \| C_t - L \|^2 \right) = \left( 1 - \frac{1}{n-1} \right)^t E \left( \| C_0 - L \|^2 \right) + \frac{n}{4}.
\]
How long does it take for each agent to converge?

Main ideas of the convergence proof (continued).

- Let $\kappa = (\#\text{red} - \#\text{blue})/100$
- For all $\delta \in (0, 1)$, $m = \left\lceil \sqrt{2n^{3/2}}/\sqrt{\delta} \right\rceil$ and for all $t \geq (n - 1) \left(5 \ln 2 + 3 \ln n - \ln \delta + \frac{2}{m-1}\right)$, we show that

$$P\{O(C_t^{(i)}) = \kappa, \text{ for all } i = 1, \ldots, n\} \geq 1 - \delta$$

- Thus the parallel convergence time to get $\kappa$ with any high probability is $O(\log n)$
- The knowledge of the population size $n$ allows us to directly derive $\#\text{red} - \#\text{blue}$
Figure: Evolution of the configuration vector for a conserved advantage \#red - \#blue equal to $3n/5$. Settings: $n = 2^{22} = 4,19 \times 10^6$. 
Figure: Evolution of the configuration vector for a conserved advantage \#red - \#blue equal to 0. Settings: $n = 2^{22} = 4 \times 10^6$. 
Figure: Evolution of the configuration vector for a conserved advantage \#red - \#blue equal to $3n/5$. Settings: $n = 2^{22} = 4, 19 \times 10^6$. 
Figure: Evolution of the configuration vector for a conserved advantage. #red - #blue equal to 0. Settings: \( n = 2^{22} = 4,19 \times 10^6 \).
Figure: Evolution of the configuration vector for a conserved advantage 
#red - #blue equal to 3n/5. Settings: n = 2^{22} = 4, 19 \times 10^6.
Figure: Evolution of the configuration vector for a conserved advantage \texttt{#red - #blue} equal to 0. Settings: \( n = 2^{22} = 4,19 \times 10^6 \).
Related works: Computation of the majority, i.e., whether red > blue

- [DV12]$^2$, [MNRS14]$^3$:
  - Four-state protocol with an expected convergence parallel time in $O(\log n)$
  - When red $\simeq$ blue, the convergence parallel time is infinite

---


Related works: Computation of the majority, i.e., whether \#red > \#blue

- [AAE07]\textsuperscript{4}, [PVV09]\textsuperscript{5}:
  - Three-state protocol with an expected convergence parallel time in $O(\log n)$
  - Only when \#red - \#blue is in $O(\sqrt{\log n})$


\textsuperscript{5} [PVV09] E. Perron, D. Vasudevan and M. Vojnovic, “Using three states for binary consensus on complete graphs”, INFOCOM, pp 2527-3435, 2009
Related works: Computation of the majority, i.e., whether \#red > \#blue

- [AGV15]^{6}:
  - (log \(n\))-state protocol with an expected convergence parallel time in \(O(\log n)\)
  - Whatever the difference between \#red and \#blue
  - The authors show that a convergence in \(O(\log n)\) interactions in expectation is a lower bound.

---

The population protocol model: A simple but powerful enough model to describe distributed systems

The exact computation power of these protocols has been determined

We have proposed simple proofs to show that the convergence time for the counting problem is logarithmic in the size of the population