Mediated Population Protocols

Othon Michail
Paul Spirakis
Ioannis Chatzigiannakis

Research Academic Computer Technology Institute
(RACTI)

July 2009

Outline I

1. Population Protocols
   - Definition
   - Stability
   - Computational Power

2. Mediated Population Protocols
   - Definition
   - The Role of Edges
   - Stability
   - Computational Power

The Systems

- A **communication graph** $G = (V, E)$.
- V: A **population** of $|V| = n$ **agents** (sensor nodes).
- E: The permissible pairwise interactions between the agents.
- Each agent is a self-contained package consisting of
  - control unit, **constant** memory, low-power wireless communication mechanism, limited power supply.
  - Agents are **passively mobile**.
  - Communicate when they come sufficiently close to each other.
- Agents are represented as **communicating finite-state machines**.

Formal Definition of Population Protocols

[Angluin, Aspnes, Diamadi, Fischer, and Peralta, PODC '04]

A PP $\mathcal{A}$ consists of

- finite **input and output alphabets** X and Y,
- finite set of **states** Q,
- **input function** $I : X \rightarrow Q$,
- **output function** $O : Q \rightarrow Y$,
- **transition function** $\delta : Q \times Q \rightarrow Q \times Q$.

$\delta(p, q) = (p', q')$ or simply $(p, q) \rightarrow (p', q')$ is called a **transition**.

Significant Properties

- **Uniformity:** Protocol descriptions are independent of the population size n.
- **Anonymity:** There is no room in the state of an agent to store a unique identifier.

Computation

- Model passive movement by an **adversary scheduler** that picks members of E.
- $C : V \rightarrow Q$, **population configuration** specifying the state of each agent.

Execution: Finite or infinite sequence $C_0, C_1, C_2, \ldots$, s.t. $C_i \rightarrow C_{i+1}$ for all i.
Fairness Formally: For all C, C' s.t. $C \rightarrow C'$, if C occurs infinitely often in the execution the same holds for C'.
Computation: Infinite fair execution.

Stable Computation

- C is **output-stable** if $O(C') = O(C)$ for all C' reachable from C (no agent changes its output).
- A PP $\mathcal{A}$ **stably computes a predicate** $p : \mathcal{X} \rightarrow \{0, 1\}$ iff for every $x \in \mathcal{X}$ and every computation that starts from $I(x)$ (initial configuration) the computation reaches an output-stable configuration C, under which all agents output the value $p(x)$.

Semilinear Predicates - Presburger Arithmetic

Semilinear Predicates

- A predicate on input assignments is **semilinear** if its support (input assignments mapped to 1) is a semilinear set.

Presburger Arithmetic [Presburger 1929]

- Arithmetic on natural numbers with addition **but not multiplication**.
- Formulas involving addition, $<$, mod-k congruence relation $\equiv_k$ for each constant k and usual logical connectives $\vee$, $\wedge$ and $\neg$.

Semilinear sets are those that can be defined in Presburger arithmetic [Ginsburg and Spanier, 1966].

Exact Characterization

Theorem

A predicate is computable in the basic population protocol model if and only if it is semilinear [Angluin et al. 2004, 2006].

Stably Computable (semilinear)

- "The number of a's is greater than 5" (i.e. $N_a > 5$).
- $(N_a = N_b) \vee (\neg(N_b > N_c))$.

Non-stably computable (non-semilinear)

- "The number of c's is the product of the number of a's and the number of b's" (i.e. $N_c = N_a \cdot N_b$).

Formal Definition of Mediated Population Protocols

[I. Chatzigiannakis, O. Michail, and P. G. Spirakis, ICALP '09]

Population V of $|V| = n$ agents forming a communication graph $G = (V, E)$. A MPP $\mathcal{A}$ consists of

- finite **input and output alphabets** X and Y,
- finite set of **agent states** Q, **agent input function** $I : X \to Q$, **agent output function** $O : Q \to Y$,
- finite set of **edge states** S, **edge input function** $\iota : X \to S$, **edge output function** $\omega : S \to Y$,
- **output instruction** r,
- (**totally ordered cost set** K, **cost function** $c : E \to K$) optional, and
- **transition function** $\delta : Q \times Q \times K \times S \to Q \times Q \times K \times S$.

Mediated Population Protocols

Transition function $\delta : Q \times Q \times S \rightarrow Q \times Q \times S$

- Assume that costs are not defined.
- When agents u_1, u_2 in states a, b, respectively, interact through (u_1, u_2) in state s then $(a, b, s) \rightarrow (a', b', s')$ is applied and
  - a goes to a',
  - b goes to b', and
  - s goes to s'.

New Assumptions about Edges...

- We assume that each edge is equipped with a buffer of $\mathcal{O}(1)$ storage capacity (independent of the population).
  - **Each pair of communicating agents shares a memory of constant size**.
- During interaction (u, v) the corresponding agents read the memory contents and update it according to δ.

Network Configurations

- A **network configuration** is a mapping $C : V \cup E \rightarrow Q \cup S$ specifying the agent state of each agent in the population and the edge state of each edge in the communication graph.

r-stability

- **Problem**: "*Given an undirected communication graph $G = (V, E)$ and a useful cost function $c : E \to K$ on the set of edges, design a protocol that will find the minimum cost edges of E*".
- **Example of instruction** r: "*Get each $e \in E$ for which $\omega(s_e) = 1$ (where s_e is the state of e)*".

r-stable network configuration C: for every C', (i) If a subgraph needs to be found, C fixes a subgraph that doesn't change in any C' reachable from C (ii) If a function has to be computed by the agents, then an r-stable configuration drops down to an agent output-stable configuration.

- **Permits protocols that search for and, eventually, find certain subgraphs of G.**

Stable Computation

Definition

A protocol $\mathcal{A}$ **stably solves** a problem Π, if for every instance I of Π and every computation of $\mathcal{A}$ on I, the network reaches an r-stable configuration C that gives the correct solution for I if interpreted according to the output instruction r. If Π is a function f to be computed we say instead that $\mathcal{A}$ **stably computes** f.

Some (Stably) Solvable Problems

- Find the **edges of minimum cost**.
- Find a **maximal matching**.
- Find a **vertex cover** which is at most $2 \cdot OPT$.
- By assuming a unique leader in the initial configuration construct the **transitive closure** of G.

MPP is stronger than PP

- Obviously, PP model is a special case of MPP model.
    - Ignore edge functions, states, costs, and instruction r to get PP.
- The edge buffers enable each pair of agents to remember a **pairwise history of up to $\mathcal{O}(1)$ interactions**.

For Example...

Assume a complete directed graph $G = (V, E)$.

- $N_c = N_a \cdot N_b$.
- Rephrase it: "**Is N_c equal to the number of links leading from agents with input a to agents with input b?**"
- This predicate is **not semilinear**, since Presburger arithmetic does not allow multiplication of variables.

$N_c = N_a \cdot N_b$ protocol

VarProduct

- $X = \{a, b, c, 0\}$,
- $Y = \{0, 1\}$,
- $Q = \{a, \dot{a}, b, c, \bar{c}, 0\}$,
- $I(x) = x$, for all $x \in X$,
- $O(a) = O(b) = O(\bar{c}) = O(0) = 1$, and $O(c) = O(\dot{a}) = 0$,
- $S = \{0, 1\}$,
- $\iota(x) = 0$, for all $x \in X$,
- r: "*If there is at least one agent with output 0, reject, else accept.*",
- δ:

$$
\begin{aligned}
(a, b, 0) &\rightarrow (\dot{a}, b, 1) \\
(c, \dot{a}, 0) &\rightarrow (\bar{c}, a, 0) \\
(\dot{a}, c, 0) &\rightarrow (a, \bar{c}, 0)
\end{aligned}
$$

Proof Sketch

- For each a the protocol tries to erase b c's.
- Each a is able to remember the b's that has already counted by marking the corresponding links.
- If the c's are less than the product then at least one $\dot{a}$ remains and if the c's are more at least one c remains. In both cases at least one agent that outputs 0 remains.
- If $N_c = N_a \cdot N_b$ then every agent eventually outputs 1.

Discussion

- A MPP **strongly stably computes** a predicate if in every computation all agents eventually agree on the correct output value.
- This is not the case for *VarProduct* (sometimes only one agent eventually gives output 0 and the answer is "reject").
- But it is easy to see that that *VarProduct* has **stabilizing states**:
  - In every computation all agents eventually stop changing their state (stronger than stabilizing outputs).
- Moreover, instruction r defines a **semilinear predicate on multisets of *VarProduct*'s states** (we can write it formally as $(N_c > 0) \vee (N_{\dot{a}} > 0)$).
- We exploit these properties to prove that with a slight modification *VarProduct* strongly stably computes $N_c = N_a \cdot N_b$.

Composition Theorem

Theorem

Any MPP $\mathcal{A}$, that stably computes a predicate p with stabilizing states in some family of directed and connected communication graphs $\mathcal{G}$, containing an instruction r that defines a semilinear predicate t on multisets of $\mathcal{A}$'s agent states, can be composed with a provably existing MPP $\mathcal{B}$, that strongly stably computes t with stabilizing inputs in $\mathcal{G}$, to give a new MPP $\mathcal{C}$ satisfying the following properties:

- *$\mathcal{C}$ is formed by the composition of $\mathcal{A}$ and $\mathcal{B}$,*
- *its input is $\mathcal{A}$'s input,*
- *its output is $\mathcal{B}$'s output, and*
- *$\mathcal{C}$ strongly stably computes p in $\mathcal{G}$.*

Non-uniform Upper Bounds

Let *DMP* be the class of predicates stably computable by the MPP model in any family of directed communication graphs.

Theorem

All predicates in DMP are also in $NSPACE(m)$.

Summary

- **Population Protocol model** was the **first step** in this widely unexplored field of societies of tiny networked artefacts.

Our research:

- **Mediated Population Protocol model**: A natural extension of the PP model gives birth to a **promising new area of research**.
- Many new directions: **finding subgraphs**, **deciding graph properties**, **optimization**...
- Moreover the MPP model is computationally stronger than the Population Protocol model.
- **Verification** is the key for applying such protocols in real, critical systems.

Thank You!