READING AND DISCUSSION

OBJECT DISTRIBUTION IN P SYSTEMS

Outline

- Motivation
- About the paper:
  - Introduction
  - Formal Definitions
  - Approaches
- Our problem:
  - Simulation algorithm
  - Ideas

Motivation

- Parallel simulation Probabilistic P systems on GPUs/CUDA.
- Probabilistic P systems:
  - Multienvironment, multicompartmental, probabilities associated to rules.
  - Rules of the form: $r:\ u\,[\ v\ ]_i^{\alpha} \xrightarrow{c_r} u'\,[\ v'\ ]_i^{\alpha'}$
  - LHS of rules can have intersections (competition for objects).

Introduction

- An Algorithm for Non-deterministic Object Distribution in P Systems and Its Implementation in Hardware
  - Van Nguyen, David Kearney, Gianpaolo Gioiosa: School of Computer and Information Science, University of South Australia,
  - Membrane Computing: 9th International Workshop, WMC 2008, Edinburgh, UK, July 28-31, 2008, Revised Selected and Invited Papers.
  - 325 - 354

Introduction

- Transition P system:

Definition 1. *A P system is a tuple*

$$\Pi = (O, \mu, w_1, \ldots, w_n, R_1, \ldots, R_m, i_0),$$

where:

(i) O *is an alphabet (i.e., a set of distinct entities) whose elements are called objects.*

(ii) μ *is the membrane structure of the particular P system; membranes are injectivelly labeled with succeeding natural numbers starting with one.*

(iii) w_i, $1 \leq i \leq m$, *are strings that represent multisets over* O *associated with each region* i.

(iv) R_i, $1 \leq i \leq m$, *are finite sets of rewriting rules (called evolution rules) over* O. *An evolution rule is of the form* $u \rightarrow v$, $u \in O^+$ *and* $v \in O_{\mathrm{tar}}^+$, *where* $O_{\mathrm{tar}} = O \times \mathrm{TAR}$, $\mathrm{TAR} = \{\mathrm{here}, \mathrm{out}\} \cup \{\mathrm{in}_j | 1 \leq j \leq m\}$.

(v) $i_0 \in \{1, 2, \ldots, m\}$ *is the label of an elementary membrane (i.e., a membrane that does not contain any other membrane), called the output membrane.*

Introduction

- Maximal Parallelism:
  - In a transition of a P system, if any reaction rule can be applied, it must be applied.
- Non-determinism:
  - When the reaction rules in a region are applied in a maximally parallel manner during a transition, there are often multiple ways in which the objects in the region can be distributed to the reaction rules.

Introduction

$$a^{13}\, b^{15}\, c^{10}$$

R_1: $a^2\, b\, c^5 \rightarrow \ldots$

R_2: $a\, b^2\, c^4 \rightarrow \ldots$

R_3: $a^3\, b^3\, c \rightarrow \ldots$

R_4: $a^4\, b^5\, c^3 \rightarrow \ldots$

(a)

R_1	R_2	R_3	R_4
0	0	0	3
0	0	1	2
0	0	3	1
0	1	0	2
0	1	2	1
0	1	4	0
0	2	2	0
1	0	2	1
1	0	3	0
1	1	1	0
2	0	0	0

(b)

Formal Definitions

- *n* reaction rules in a region:

$$
\begin{aligned}
R_1&: o_1^{a_{11}} o_2^{a_{21}} \ldots o_m^{a_{m1}} \rightarrow \ldots \\
R_2&: o_1^{a_{12}} o_2^{a_{22}} \ldots o_m^{a_{m2}} \rightarrow \ldots \\
&\ldots \\
R_n&: o_1^{a_{1n}} o_2^{a_{2n}} \ldots o_m^{a_{mn}} \rightarrow \ldots,
\end{aligned}
$$

- *m* objects in a multiset: $o_1^{b_1} o_2^{b_2} \ldots o_m^{b_m}$
- Instances of reaction rules: $x_1, x_2, \ldots, x_n$
- It must be satisfied:

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &\leq b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &\leq b_2 \\
&\ldots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &\leq b_m
\end{aligned}
$$

Formal Definitions

- Maximal parallelism:
  - Consider solution: $s = (s_1,\ s_2,\ \ldots,\ s_n)$
  - *s* corresponds to maximal application i.o.i.:
    - Solution $v = (v_1, v_2, \ldots, v_n)$

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &\le b'_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &\le b'_2 \\
&\ldots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &\le b'_m,
\end{aligned}
$$

Formal Definitions

- Maximal parallelism:

where

$$
\begin{aligned}
b'_1 &= b_1 - a_{11}s_1 - a_{12}s_2 - \ldots - a_{1n}s_n \\
b'_2 &= b_2 - a_{21}s_1 - a_{22}s_2 - \ldots - a_{2n}s_n \\
b'_m &= b_m - a_{m1}s_1 - a_{m2}s_2 - \ldots - a_{mn}s_n
\end{aligned}
$$

is such that v is the zero vector.

Formal Definitions

- Non-determinism:

$$\text{There are } p \geq 0 \text{ possible values for } s$$

Approaches

- Indirect approaches:
  - Algorithms that consider both non-solutions and solutions as it navigates the space of possible solutions.
  - **Indirect straightforward approach:** To simply enumerate all the possible solutions and pick at random.
  - **Incremental approach:** The distribution of objects is accomplished in rounds, randomly choosing rules from a pool.

Approaches

- Direct approaches:
  - Algorithms that consider only solutions as it navigates the space of possible solutions.
  - **Direct straightforward approach:** All the solutions are give as input, and one of them is selected at random.
  - **Direct non-deterministic distribution algorithm (DND algorithm):** Performs the distribution in one step (with a possible adjustment step). Based on two phases.

DND Algorithm

a_{11}	a_{12}	...	a_{1n}
a_{21}	a_{22}	...	a_{2n}
...	...	...	...
a_{m1}	a_{m2}	...	a_{mn}

A

c_1
c_2
...
c_m

C

x_1
x_2
...
x_n

X

b_{11}	b_{12}	...	b_{1n}
b_{21}	b_{22}	...	b_{2n}
...	...	...	...
b_{m1}	b_{m2}	...	b_{mn}

B

---

procedure obtainASolutionnon-deterministically (

m: number of object types required by a reaction rule
n: number of reaction rules in the region
A: an $m \times n$ matrix (with initially unmarked columns) used to store the coefficients of the linear system
B: an initially empty $m \times n$ matrix that contains results of calculations carried out on A
C: an $m \times 1$ matrix that contains the RHS constants of the linear system
X: an $n \times 1$ matrix used to store the solution
V: an $m \times 1$ matrix used to store accumulated sums used in the calculation of values to be stored in B
Z: a list of integer labels for columns of A (ordered according to the order in which the columns of A are processed)

DND Algorithm

//Forward phase

1. **for** $u = 1$ to n

2. Randomly select a column p from all the unmarked columns in A ($1 \leq p \leq n$).

3. Add p to Z.

4. **if** B is empty

5. **let** q be the minimum value of all c_i/a_{ip} ($1 \leq i \leq m$).

6. **else**

7. **let** q be the minimum value of all $b_{i(u\text{-}1)}/a_{ip}$ ($1 \leq i \leq m$).

8. **if** p is the only unmarked column in A

9. **if** q is an integer

10. Set $x_p = q$. **End procedure**.

11. **else**

12. Set $x_p = \lfloor q \rfloor$. Go to 22.

13. **else**

14. **if** $q = 0$

15. Set $x_p = q$ and mark x_p as final.

16. **else**

17. Randomly select $r \in \{0, 1, \ldots, \lfloor q \rfloor\}$ and set $x_p = r$.

18. For all i ($1 \leq i \leq m$), set $v_i = v_i + r a_{ip}$.

19. For all i ($1 \leq i \leq m$), set $b_{iu} = c_i - v_i$.

20. Mark column p in A.

21. **end for**

DND Algorithm

//Backward phase

Reset V and set $v_i = x_{Z(n)} a_{iZ(n)}$ for all i $(1 \le i \le m)$.

for $s = n - 1$ to 1

    if $x_{Z(s)}$ is not marked as final

        if $s = 1$

            let q' be the minimum value of all $(c_i - v_i)/\ a_{iZ(1)}$ $(1 \le i \le m)$.

        else

            let q' be the minimum value of all $(b_{i(s-1)} - v_i)/\ a_{iZ(s)}$ $(1 \le i \le m)$.

        if $x_{Z(s)} \ne q'$ set $x_{Z(s)} = \lfloor q' \rfloor$.

        For all i $(1 \le i \le m)$, set $v_i = v_i + \lfloor q' \rfloor\ a_{iZ(s)}$.

    end if

end for

DND Algorithm

- Time complexity: O(n), n=number of rules
- Space complexity: O(2mn+n), m=number of objects, n=number of rules.
- The algorithm is verified, and a hardware implementation is defined.

Our problem

- Probabilistic P systems:
  - Multienvironment, multicompartmental, probabilities associated to rules.
  - Rules of the form: $r: \; u \, [\; v \;]_i^{\alpha} \xrightarrow{c_r} u' \, [\; v' \;]_i^{\alpha'}$
  - Note that:
    - Objects of different regions collaborate
    - Consider all the regions as a superregion where to apply this algorithm
    - For the model of the zebra mussel ecosystem, the algorithm requires more than 3Gbytes, and all the rule selection stage is performed in sequential mode.

Ideas

- Identify rules with intersection in the RHS, and apply only for them this algorithm.
- Considering the caractheristics of the current models, to develop a specific algorithm based on the DND.

Thanks for listening