CDMTCS Research Report Series

Using Membrane Systems to Solve the Bounded Fanout Broadcast Problem

Michael J. Dinneen
Yun-Bum Kim

Department of Computer Science,
University of Auckland,
Auckland, New Zealand

CDMTCS-467
July 2014

Centre for Discrete Mathematics and
Theoretical Computer Science

Using Membrane Systems to Solve the Bounded Fanout Broadcast Problem

Michael J. Dinneen and Yun-Bum Kim

Department of Computer Science, University of Auckland,
Private Bag 92019, Auckland, New Zealand
mjd@cs.auckland.ac.nz, tkim021@aucklanduni.ac.nz

July 28, 2014

Abstract

Broadcasting is the information distribution process in a communication network, which aims to inform all network nodes with a unique message, initially held by a subset of nodes called originators. This paper considers a decision problem that asks if it is possible to inform all nodes within t time units. This paper presents a non-deterministic solution, implemented with a bio-inspired distributed and parallel computational model called membrane systems, which decides in $t+1$ steps.

Keywords: P systems, broadcast, fanout, communication network.

1 Introduction

For a given communication network $G = (V, E)$, broadcasting from node $v \in V$ is the process of distributing information from v to every other node, under the following constraints: (i) messages are exchanged between neighboring nodes, (ii) each message exchange takes one time unit, (iii) each node can exchange messages with up to $f \geq 1$ neighbors in one time unit. The problem is to design a messaging protocol that informs all network nodes from a starting set of vertices with the unique message within a deadline. This problem, a variant to the *Minimum Broadcast Time Problem* [3, 2], is formulated next in Problem 1.

Problem 1. The Bounded Fanout Broadcast Problem
Instance: graph $G = (V, E)$, subset $V_0 \subseteq V$ called *originators*, a positive integer f called *fanout*, a positive integer t called *deadline*.
Question: Is there a sequence of sets $V_0, E_1, V_1, \ldots, E_t, V_t$, such that each $V_i \subseteq V$, each $E_i \subseteq E$, $V_t = V$, and, for $1 \leq i \leq t$,

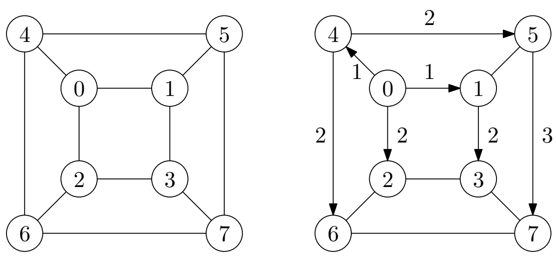
1. $V_i = V_{i-1} \cup \{v \mid (u, v) \in E_i\}$,

2. each edge in E_i has an endpoint in both V_{i-1} and $V_i \setminus V_{i-1}$,

3. each vertex in V_{i-1} is incident to at most f edges in E_i,

4. each vertex in $V_i \setminus V_{i-1}$ is incident to at most 1 edge in E_i.

The set of edges E_i, $1 \leq i \leq t$, satisying the constraints of Problem 1 is considered to be a *broadcast tree (protocol)* of time t for a graph G. Usually the set of originators is a single source vertex $v \in V$. We say the *fanout f broadcast time* of G originating at v, denoted $BT_f(G, v)$, is the smallest value t such that there is corresponding broadcast tree of time t.



$V_0 = \{0\}$
$E_1 = \{(0,1),(0,4)\}$
$V_1 = \{0,1,4\}$
$E_2 = \{(0,2),(1,3),(4,5),(4,6)\}$
$V_2 = \{0,1,2,3,4,5,6\}$
$E_3 = \{(5,7)\}$
$V_3 = \{0,1,2,3,4,5,6,7\}$

Figure 1: **Left:** A connected graph. **Center:** A graph that shows the time step in which nodes have been informed (indicated with edge labels) during broadcasting from node 0 with fanout $f = 2$. **Right:** The sequence of sets $V_0, E_1, V_1, E_2, V_2, E_3, V_3$ corresponding to the graph shown in the center.

The main contribution in this paper is to present a non-deterministic[1] solution to the bounded fanout broadcast problem using a computing model called membrane system. Membrane systems [6, 7] (also known as P systems) are distributed and parallel computing model, inspired by the structure and function of living cells. A membrane system consists of a network of (multiset processing) computing units called membranes. Each membrane contains a multiset of symbols and is associated with a set of multiset processing rules.

This paper is organized as follows. Section 2 recalls several key mathematical concepts that are used in this paper. Section 3 presents the definition of a membrane system used in this paper. Section 4 presents the details of constructing a membrane system that solves the bounded fanout broadcast problem for a given instance. Finally, Section 5 summarizes this paper and provides some open problems.

2 Preliminaries

This section covers several key mathematical concepts that are used in this paper, such as sets, strings, multisets and graphs.

An *alphabet* is a finite non-empty set with elements called *symbols*. A *string over* alphabet O is a finite sequence of symbols from O. The set of all strings over O is

[1] each informed node non-deterministically selects uninformed neighbors

denoted by O^*. The length of a string $x \in O^*$, denoted by $|x|$, is the number of symbols in x. The number of occurrences of a symbol $o \in O$ in a string x over O is denoted by $|x|_o$. The *empty string* is denoted by λ.

A *multiset* is a set with multiplicities associated with its elements. A set that contains the distinct elements of a multiset v is denoted by $\mathtt{distinct}(v)$. The empty string or multiset is represented by λ. The *size* of a multiset v is denoted by $|v|$. The *multiplicity* of an element x in a multiset v is denoted by $|v|_x$. We say that a multiset v is *included* in a multiset w, denoted by $w \subseteq v$, if, for all $o \in O$, $|w|_o \leq |v|_o$. The *union* of multisets v and w, denoted by $v \cup w$, is a multiset x, such that, for all $o \in O$, $|x|_o = |v|_o + |w|_o$. The *difference* of multisets v and w, denoted by $v - w$, is a multiset x, such that, for all $o \in O$, $|x|_o = \max(|v|_o - |w|_o, 0)$.

A (binary) *relation* R over two sets X and Y is a subset of their Cartesian product, $R \subseteq X \times Y$. For $A \subseteq X$ and $B \subseteq Y$, we set $R(A) = \{y \in Y \mid \exists x \in A, (x, y) \in R\}$, $R^{-1}(B) = \{x \in X \mid \exists y \in B, (x, y) \in R\}$.

A *graph* is an ordered pair (V, E), where V is a finite set of elements called nodes and E is a set of unordered pairs of V called edges. A *path* of length $n-1$ is a sequence of n nodes, $v_1, v_2, \ldots, v_n$, such that $\{(v_1, v_2), \ldots, (v_{n-1}, v_n)\} \subseteq E$. The *diameter* of G, denoted by $\mathtt{dia}(G)$, is the maximum of the lengths of shortest paths between every pair of nodes of G.

A *directed graph* (digraph) is a pair (V, A), where V is a finite set of elements called nodes and A is a set of an ordered pair of V called *arcs*. Given a digraph $D = (V, A)$, for $v \in V$, the *parents* of v are $A^{-1}(v) = A^{-1}(\{v\})$ and the *children* of v are $A(v) = A(\{v\})$.

3 Membrane systems

Membrane systems (also known as P systems) are distributed and parallel computing model. A membrane system consists of a network of (multiset processing) computing units called membranes. Each membrane contains a multiset of symbols and is associated with a set of multiset processing rules. Several P system models [5, 4, 1] have been introduced, inspired from various features of living cells, that provide new ways to process information and solve the computational problems of interest. A membrane system model used in this paper has the form $\Pi = (O, Q, K, R, \Delta)$, where

1. O is a finite non-empty alphabet of *symbols*.

2. Q is a finite set of *states*.

3. $K = \{\mu_1, \mu_2, \ldots, \mu_n \mid n \in \mathbb{N}^+\}$ is a finite set of *membranes*. Each membrane $\mu_i \in K$ is of the form $\mu_i = (s_i, w_i)$, where
   - $s_i \in Q$ denotes the *current state* of μ_i,
   - $w_i \in O^*$ denotes the *current content* of μ_i.

4. R is a set of *evolution rules*, where an evolution rule $r \in R$ has the form:

$$j \ s \ u \rightarrow_\alpha s' \ v \ w \ x$$

- $\alpha \in \{\texttt{min}, \texttt{max}\}$ is a *rewriting* operator of r,
- $j \in \mathbb{N}$ is the *priority* of r, where the lower value j indicates higher priority,
- $s, s' \in Q$, where s is the *start state* and s' is the *target state* of r,
- $u \in O^+$,
- $v \in (O \times \tau)^*$, where $\tau \in \{\odot, \uparrow, \downarrow, \updownarrow\}$ is a *target indicator*. Note that, $(o, \odot) \in v$, $o \in O$, is abbreviated to o,

5. Δ is an irreflexive and asymmetric relation on K, representing a set of arcs between membranes with bidirectional communication capabilities.

A *configuration* of system Π of order n is $(s_1, w_1, s_2, w_2, \ldots, s_n, w_n)$, where, for $1 \leq i \leq n$, s_i and w_i correspond to the current state and content of membrane σ_i, respectively. Consider two configurations of system Π, C' and C''. A *transition* in system Π is a transformation from C' to C'' in one time unit, denoted by $C' \Rightarrow C''$, such that C'' is obtained from C'. A transition $C' \Rightarrow C''$ consists of two substeps (substep 1 and substep 2). All membranes simultaneously perform substep 2, after every membrane has finished substep 1.

- **Substep 1:** Each membrane μ_i, $1 \leq i \leq n$, finds a maximal multiset of evolution rules, M_i, as described in Definitions 2 and 3.

- **Substep 2:** Each membrane μ_i, $1 \leq i \leq n$, executes a multiset of evolution rules found in substep 1, M_i, as described in Definition 4.

System Π *halts*, if it reaches a configuration (called the halting configuration), where no evolution rule can be applied to the existing symbols inside all membranes. The *computational results* of a halted system are the multiplicities of symbols present in the membranes of the system.

Definition 2. Given a multiset $w \in O^*$ and an evolution rule $r \in R$, where $\texttt{LHS}(r) \subseteq w$, the number of applications of r over w is

$$\texttt{apply}(r, w) = \begin{cases} 1 & \text{if } \texttt{rewrite}(r) = \texttt{min}, \\ |w|_{\texttt{LHS}(r)} & \text{if } \texttt{rewrite}(r) = \texttt{max}. \end{cases}$$

Definition 3. For membrane μ_i, in state s_i with content w_i and a set of evolution rules R_i, a maximal multiset of rules, M_i, is obtained by the procedure below.

> **Input:** a set of evolution rules R_i and a multiset $w := w_i$.
> **Output:** a maximal multiset M_i.
> $M_i := \emptyset$
> **for each** $r_j \in R_i$ with $\texttt{source}(r_j) = s_i$, $1 \leq j \leq |R_i|$ (by priority order)
>   **if** $(M_i = \emptyset \;||\; \forall r_k \in M_i\ (\texttt{dest}(r_j) = \texttt{dest}(r_k)))$ **then**
>   **if** $(\texttt{LHS}(r_j) \subseteq w$ **then**
> $m := \texttt{apply}(r_j, w)$
> $M_i := M_i \cup \{r_j^m\}$

$w := w - \texttt{LHS}(r_j)^m$

endif

endif

endfor

Definition 4. For each membrane μ_i, $1 \le i \le n$, consider a maximal multiset of evolution rules, M_i, found according to Definition 3. For membrane μ_i with the current content w_i, multisets U_i, V_i, $V_i^{\downarrow}$, $V_i^{\uparrow}$ and $V_i^{\updownarrow}$, for each $\mu_k \in \Delta(i) \cup \Delta^{-1}(i)$, are defined as follow:

- $U_i = \bigcup_{r_j \in M_i} \texttt{LHS}(r_j)$, denotes the multiset that will be consumed from w_i.
- $V_i = \bigcup_{r_j \in M_i} \bigcup_{(o,\odot) \in \texttt{RHS}(r_j)} \{o\}$, denotes the multiset that will be produced and added to w_i.
- $V_i^{\downarrow} = \bigcup_{r_j \in M_i} \bigcup_{(o,\downarrow) \in \texttt{RHS}(r_j)} \{o\}$, denotes the multiset that will be sent to each $\mu_k \in \Delta(i)$.
- $V_i^{\uparrow} = \bigcup_{r_j \in M_i} \bigcup_{(o,\uparrow) \in \texttt{RHS}(r_j)} \{o\}$, denotes the multiset that will be sent to each $\mu_k \in \Delta^{-1}(i)$.
- $V_i^{\updownarrow} = \bigcup_{r_j \in M_i} \bigcup_{(o,\updownarrow) \in \texttt{RHS}(r_j)} \{o\}$, denotes the multiset that will be sent to each $\mu_k \in \Delta(i) \cup \Delta^{-1}(i)$.

For each membrane μ_i in state s_i with content w_i:

- If $M_i = \emptyset$, then μ_i remains in state s_i with content w_i.
- Otherwise, μ_i transforms:
  - its current state to $s_i = \texttt{dest}(r_f)$, where $r_f \in M_i$.
  - its current content w_i to w_i', where

$$w_i' = w_i - U_i \;\cup\; V_i \;\cup \bigcup_{f \in \Delta^{-1}(i)} V_f^{\downarrow} \;\cup\; \bigcup_{g \in \Delta(i)} V_g^{\uparrow} \;\cup \bigcup_{h \in \Delta(i) \cup \Delta^{-1}(i)} V_h^{\updownarrow}$$

4 Non-deterministic P systems solutions

This section presents P system Π that correspond to a non-deterministic solution to the bounded fanout broadcast problem of Problem 1. A trace of system Π for the example of Figure 1 is given in Section 4.4.

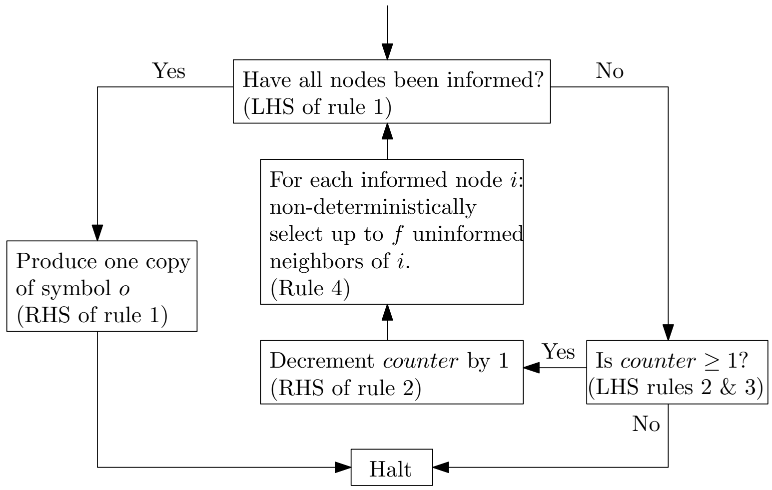


Figure 2: Procedure for μ to determine if all nodes can be informed within t steps from nodes of V_0. Initially, nodes of V_0 are marked as "informed" and every other node is marked as "uninformed". Variable *counter* has an initial value of input parameter t.

4.1 Overview of system Π

System Π consists of one membrane, labeled μ, that determines if every node can be informed within t steps from nodes of V_0, using the procedure illustrated in Figure 2. Activities and decisions indicated inside boxes of the procedure are accompanied by the corresponding evolution rules specified in Section 4.2.

As illustrated in Figure 2, μ produces one copy of symbol o if every node can be informed within t steps. The final configuration of a halted system Π can be interpreted, with respect to Problem 1, as follows:

- If μ ends with one copy of symbol o, then the answer is "Yes".
- Otherwise, the answer is "No".

4.2 Specification of system Π

Specification of system Π described earlier is (O, Q, R, K, Δ), where

1. $O = \{v_i, u_i, e_{i,j}, h, o \mid i, j \in \{1, 2, \ldots, n\}\}$.
    - Symbols $e_{i,j}$ and $e_{j,i}$ represent edge $(i, j) \in E$.
    - Symbols v_i and u_i represent the "informed" and "uninformed" status of node $i \in V$, respectively.
    - Multiplicity of symbol v_i represents the fanout parameter f.
    - Recall variable *counter* of Figure 2, which has an initial value of input parameter t. Multiplicity of symbol h corresponds to value *counter* $+ 1$.
    - Symbol o represents "Yes-output", i.e. every node can be informed within in t steps.

2. $Q = \{s_0, s_1, s_2\}$, where

   - s_0 represents an active state where informed nodes non-deterministically select up to f uninformed nodes.
   - s_1 represents a halt state where all nodes could not be informed within t steps.
   - s_2 represents a halt state where every node is informed within t steps.

3. R corresponds to the following rules. The task each rule undertakes is indicated in Figure 2.

   1. $s_0\ v_1^f\ v_2^f\ \ldots\ v_n^f \rightarrow_{\texttt{min}} s_2\ o$
   2. $s_0\ h\ h \rightarrow_{\texttt{min}} s_0\ h$
   3. $s_0\ h \rightarrow_{\texttt{min}} s_1\ h$
   4. $s_0\ v_i\ e_{i,j}\ e_{j,i}\ u_j \rightarrow_{\texttt{min}} s_0\ v_i\ v_j^f$

4. $K = \{\mu\}$, where μ has the initial form of $(s_0, V_K \cup U_K \cup E_K \cup h^{t+1})$, where

   - $V_K = \{{v_j}^f \mid j \in \{V_0\}\}$,
   - $U_K = \{u_j \mid j \in \{1, 2, \ldots, n\} \setminus \{V_0\}\}$,
   - $E_K = \{e_{i,j}, e_{j,i} \mid (i, j) \in E\}$.

5. $\Delta = \emptyset$.

4.3 Analysis of system Π

Propositions 5 and 6 demonstrate the correctness of construction of system Π for solving the Problem 1. The run-time complexity of system Π is indicated in Proposition 7.

Proposition 5. Using rule 4, each informed node non-deterministically selects f uninformed neighbors repeatedly, if any, and marks them as "informed".

Proof. Each copy of symbol v_i is used to find one uninformed neighbor, if any, as follows. If symbols v_i, u_j, $e_{i,j}$ and $e_{j,i}$ are available (i.e. node i is visited, node j is unvisited and nodes i and j are neighbors), then rule 4 rewrites symbol u_j into f copies of symbol v_j (i.e. transforms the status of node j from "uninformed" to "informed"). Every copy of symbol v_i is preserved, such that node i can select up to f uninformed neighbors in the future repeatedly, if necessary. □

Proposition 6. Membrane μ replicates the the procedure of Figure 2.

Proof. We show that the evolution rules of R, which govern the behavior of μ resemble the procedure of Figure 2. Membrane μ starts from state s_0. Membrane μ in state s_0 finds and executes rules in each step as follows:

- Due to the rule priority, rule 1 is the first rule checked by μ. Rule 1 inspects whether every node is informed by requiring multiset $\{v_i^f \mid 1 \leq i \leq n\}$. If μ meets this requirement, rule 1 is executed, which prompts μ to produce one copy of symbol o and halt by entering state s_2.

- Rule 2 is the next rule checked by μ, given that μ does not contain multiset $\{v_i^f \mid 1 \leq i \leq n\}$ (i.e. not every node is informed). Rule 2 inspects the condition "*counter* $\geq$ 1?" by requiring multiset $\{hh\}$. If μ contains $\{hh\}$, rule 2 is executed, which prompts μ to consume one copy of symbol h (i.e. decrement *counter* by 1) and remain in state s_0 such that μ can check through rules of R in the next step.

- Rule 3 is the rule executed by μ, given that μ does not satisfy the requirements of rules 1 and 2, i.e. not every node is informed and *counter* $= 0$. Executing rule 3 prompts μ to halt by entering state s_1.

- Rule 4 can be executed in parallel with rule 2 in one step, since these rules have the same target state of s_0. As described in Proposition 5, rule 4 enables each informed node to non-deterministically select up to f uninformed neighbors.

The manner in which rules 1, 2, 3 and 4 are selected, and the results these rules produce resemble the procedure of Figure 2. Thus, μ replicates the procedure of Figure 2. □

Proposition 7. System Π takes at most $t + 1$ steps.

Proof. In each step, μ executes (i) rule 1, (ii) rules 2 and 4, or (iii) rule 3. The maximum number of steps rules 2 and 4 can be executed is t. If all nodes have been informed in $t' \leq t$ steps, then μ halts at step $t' + 1$ by executing rule 1. Otherwise, μ halts at step $t' + 1$ by executing rule 3. □

4.4 Example - an evolution trace of system Π

The table below illustrates an evolution trace of system Π for the instance: G is the graph shown in Figure 1 (Left), initiators $V_0 = \{0\}$, fanout $f = 2$ and deadline $t = 3$. The order in which nodes are informed in the trace below corresponds to the sequence given in Figure 1 (Right). The table indicates the state and content of membrane μ in each step. The content column is divided into five sub-columns that respectively indicate (i) "edge" symbols, (ii) "counter" symbol, (iii) "unvisited node" symbols, (iv) "visited node" symbols and (v) "Yes-output" symbol.

Step	State	Content				
0	s_0	$e_{0,1}$ $e_{0,2}$ $e_{0,4}$ $e_{1,0}$ $e_{1,3}$ $e_{1,5}$ $e_{2,0}$ $e_{2,3}$ $e_{2,6}$ $e_{3,1}$ $e_{3,2}$ $e_{3,7}$ $e_{4,0}$ $e_{4,5}$ $e_{4,6}$ $e_{5,1}$ $e_{5,4}$ $e_{5,7}$ $e_{6,2}$ $e_{6,4}$ $e_{6,7}$ $e_{7,3}$ $e_{7,5}$ $e_{7,6}$	h^4	u_1 u_2 u_3 u_4 u_5 u_6 u_7	v_0^2	
1	s_0	$e_{0,2}$ $e_{1,3}$ $e_{1,5}$ $e_{2,0}$ $e_{2,3}$ $e_{2,6}$ $e_{3,1}$ $e_{3,2}$ $e_{3,7}$ $e_{4,5}$ $e_{4,6}$ $e_{5,1}$ $e_{5,4}$ $e_{5,7}$ $e_{6,2}$ $e_{6,4}$ $e_{6,7}$ $e_{7,3}$ $e_{7,5}$ $e_{7,6}$	h^3	u_2 u_3 u_5 u_6 u_7	v_0^2 v_1^2 v_4^2	
2	s_0	$e_{1,5}$ $e_{2,3}$ $e_{2,6}$ $e_{3,2}$ $e_{3,7}$ $e_{5,1}$ $e_{5,7}$ $e_{6,2}$ $e_{6,7}$ $e_{7,3}$ $e_{7,5}$ $e_{7,6}$	h^2	u_7	v_0^2 v_1^2 v_2^2 v_3^2 v_4^2 v_5^2 v_6^2	
3	s_0	$e_{1,5}$ $e_{2,3}$ $e_{2,6}$ $e_{3,2}$ $e_{3,7}$ $e_{5,1}$ $e_{6,2}$ $e_{6,7}$ $e_{7,3}$ $e_{7,6}$	h		v_0^2 v_1^2 v_2^2 v_3^2 v_4^2 v_5^2 v_6^2 v_7^2	
4	s_2	$e_{1,5}$ $e_{2,3}$ $e_{2,6}$ $e_{3,2}$ $e_{3,7}$ $e_{5,1}$ $e_{6,2}$ $e_{6,7}$ $e_{7,3}$ $e_{7,6}$	h			o

4.5 Remark

There are several variants to this bounded fanout broadcast problem. One of the variants is to compute the fanout f broadcast time of a graph $G = (V, E)$, defined $BT_f(G) = \max_{v \in V} BT_f(G, v)$, where the broadcast time of an originator, $BT(G, f, v)$, was defined just after Problem 1.

An overview of P system Π' that can solve this global broadcast problem is as follows. Assume that for the input graph G, $V = \{v_1, v_2, \ldots, v_n\}$. System Π' consists of $n+1$ membranes, labeled $\mu_{skin}, \mu_{v_1}, \mu_{v_2}, \ldots, \mu_{v_n}$, which are arranged in a rooted tree structure of Figure 3.

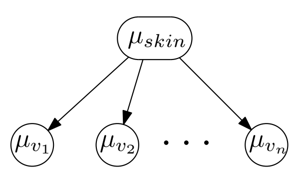


Figure 3: The membrane structure of system Π'.

Membrane μ_{v_i}, $1 \le i \le n$, covers the instance $V_0 = \{v_i\}$ by determining if node v_i can inform every node within t steps. Membrane μ_{v_i} uses the procedure illustrated in Figure 2 with the following difference: instead of producing one copy of symbol o locally, μ_{v_i} sends up one copy of symbol o to membrane μ_{skin}, i.e. replace rule $s_0\ v_1^f\ v_2^f\ \ldots\ v_n^f \rightarrow_{\texttt{min}} s_2\ o$ with $s_0\ v_1^f\ v_2^f\ \ldots\ v_n^f \rightarrow_{\texttt{min}} s_2\ (o, \uparrow)$. The final configuration of a halted system Π' can be interpreted as follows:

- If μ_{skin} ends with n copies of symbol o, then the answer is "Yes".
- Otherwise, the answer is "No".

5 Conclusions

In this paper, we studied a communication networks problem, called the bounded fanout broadcast problem, that asks: is it possible to informed all network nodes within a specified deadline, under a communication constraint that limits the number of neighbors each node can communicate simultaneously?

We designed our solution to this decision problem using membrane systems that decides within $t+1$ steps, where t denotes the deadline. Future work include two natural optimization problems: (i) find smallest fanout f when deadline t is fixed, and (ii) find smallest t when f is fixed.

Acknowledgment

We want to thank Cristian Calude for the suggestion to work on this topic. This paper was supported by the Quantum Computing Research Initiatives at Lockheed Martin.

References

[1] M. J. Dinneen, Y.-B. Kim, and R. Nicolescu. A faster P solution for the Byzantine agreement problem. In M. Gheorghe, T. Hinze, and G. Păun, editors, *Conference on Membrane Computing*, volume 6501 of *Lecture Notes in Computer Science*, pages 175–197. Springer-Verlag, Berlin Heidelberg, 2010.

[2] M. J. Dinneen, G. Pritchard, and M. C. Wilson. Degree- and time- constrained broadcast networks. *Networks*, 39(3):121–129, Mar. 2002.

[3] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[4] M. Ionescu, G. Păun, and T. Yokomori. Spiking neural P systems. *Fundam. Inform.*, 71(2-3):279–308, 2006.

[5] C. Martín-Vide, G. Păun, J. Pazos, and A. Rodríguez-Patón. Tissue P systems. *Theor. Comput. Sci.*, 296(2):295–326, 2003.

[6] G. Păun. *Membrane Computing: An Introduction*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

[7] G. Păun, G. Rozenberg, and A. Salomaa. *The Oxford Handbook of Membrane Computing*. Oxford University Press, Inc., New York, NY, USA, 2010.