Tuning P Systems for Solving the Broadcasting Problem

Raluca Lefticaru[1], Florentin Ipate[1], Marian Gheorghe[1,2], Gexiang Zhang[3]

[1] Department of Computer Science and Mathematics
University of Pitesti, Romania
Str. Targu din Vale 1, 110040 Pitesti, Romania
raluca.lefticaru@gmail.com, florentin.ipate@ifsoft.ro
[2] Department of Computer Science
The University of Sheffield
Regent Court, Portobello Street, Sheffield S1 4DP, UK
m.gheorghe@dcs.shef.ac.uk
[3] School of Electrical Engineering
Southwest Jiaotong University
Chengdu, 610031, P.R. China
zhgxdylan@126.com

Summary. P systems are employed in various contexts to specify or model different problems. In certain cases the system is not entirely known. In this paper we will consider the broadcasting algorithm and present a method to determine the format of the rules of the P system utilised to specify the algorithm.

1 Introduction

P systems (also called membrane systems) represent a class of parallel and distributed computing devices which are inspired by the structure and the functioning of living cells [10]. The model has been used for theoretical investigations as well as a vehicle to represent different problems from various domains.

With very few exceptions, [13, 5, 3, 4], in all previous studies the systems considered have been fully specified. There are situations when some components of a model are not known or maybe available in certain contexts and circumstances.

In the vast majority of cases, the P system rules act either within compartments or between those that share the same neighbourhood. There are only few situations (for instance, P systems with gemmation [1]) when rules of a compartment transfer objects from their current position to a destination that might be far away from their place.

In this paper we study the broadcasting algorithm defined in a P system framework [6], by considering a number of variants of P systems. We will study the

dependencies between the format of the rules in each compartment and the number of its neighbours, as well as a method to automatically generate the rules in each compartment depending on the number of neighbours. This problem is also important in the context of P systems where compartments are added to or removed from them. The structure of a system can be changed either by operations belonging to the system, like in the case of P systems with active membranes, or by external means, but this aspect is not considered in this paper.

2 Basic concepts

A P system is a computational model, inspired by the functioning and structure of the living cell. The cell-like P systems [12] consist of: (i) a hierarchical arrangement of *membranes*, embedded in the skin membrane, the one which separates the system from its environment; (ii) *objects* occurring inside the regions delimited by membranes, coding complex chemical molecules or compounds; and (iii) *rules* assigned to the regions of the membrane structure, acting upon the objects inside and the regions themselves. A membrane without any membrane inside is called an elementary one. Each membrane defines a region. Each region contains, apart from zero or many membranes, a multiset of objects and a set, in this paper, of transformation and communication rules.

A configuration of a P system is represented by the current membrane structure and the multisets of objects occurring in each region. The system will go from one configuration to a new one by applying the rules in a non-deterministic and maximally parallel manner, i.e., at each step, in each membrane it is applied a maximal multiset of rules. The system will halt when no more rules are available to be applied. Usually, the result of the computation is obtained in a specified component of the system, called the output region.

In what follows a basic P system using transformation and communication rules is formally defined. For more details look at [12], [11].

Definition 1. *A P system is a construct*

$$\Pi = (V, \mu, M_1, \ldots, M_m, R_1, \ldots, R_m, i_0),$$

where

- V *is an alphabet; its elements are called objects;*
- μ *is a membrane structure consisting of* m *membranes, with the membranes and the regions labelled in a one-to-one manner with elements of a given set* Λ*, usually, the set* $\{1, \cdots, m\}$*;* m *is called the degree of* Π*;*
- M_i*,* $1 \leq i \leq m$*, are strings which represent multisets over* V *associated with the regions of* μ*;*
- R_i*,* $1 \leq i \leq m$*, are transformation-communication rules associated with the regions of* μ*; each rule of* R_i *has the the form* $x \rightarrow y$*, where* x *is a non-empty multiset over* V*, and* y *defines a multiset over* $\{a_j | a \in V, j \in \{here, out, 1, \cdots, m\}\}$

(a_{here} means a remains in the current region, i; subsequently here will be ignored; a_{out} indicates that a has to go out of i to the outer region; a_j, $1 \leq j \leq m$, shows that a goes to the region j that must be directly contained by the current membrane); applying a rule means replacing x by y and following the target indications;

- *i_0 is a number between 1 and m which specifies the output membrane of Π.*

When a target indication, t, occurs more than once in a sequence, i.e., $a_t^1 \cdots a_t^h$ then the following shortcut notation $(a^1 \cdots a^h)_t$ is used. A P system provides a suitable framework for distributed parallel computation that develops in steps. Indeed, any computation starts by processing the initial multisets, w_i, and then in each step the rules associated to each region are applied in a non-deterministic and maximally parallel manner. The result of a computation, a multiset of simple objects, is obtained in region i_0. We notice that the rules presented above combine both transformation and communication, being responsible for evolving the objects and transferring them to regions according to specified targets. We will consider specific contexts for applying some of these rules, namely promoters and inhibitors [2]. *Promoters* are used to formalize the reaction enhancing; *inhibitors* have reaction prohibiting roles for various substances (molecules) present in cells [2].

3 Broadcasting through a P system

Broadcasting messages to the nodes of a network occurs in various communications and is well-studied for different network topologies, message lengths, transmission constraints. The problem is also formulated in the context of a basic P system and its complexity has been studied [6]. A basic broadcasting problem consists in sending a message from a node of a network to all the other nodes without revisiting them. In a P system environment it involves sending the message through the tree structure of the P system. The broadcasting algorithm for P systems [6] does not discuss the format of the rules that may lead to various types of P systems and, more important, specific complexity aspects of the communication processes involved.

We will first present various variants of P systems and analyse complexity aspects related to the communication processes that occur and the dependencies between the format of the rules in a compartment and the number of its neighbours.

The broadcasting problem is presented through the P system having the membrane structure given by the tree structure in Figure 1(a) where the message will start from membrane 9. According to the broadcasting principle, illustrated in [6], from each membrane, or node of the tree, the message is sent one level up, to its parent membrane, and to all its directly contained membranes. Initially the message from membrane 9 is sent to $6, 11, 12$. In the following step from these compartments the messages are sent to $3, 10$, $15, 16$, respectively. Please note that from the membranes $15, 16, 12$ the message does no longer travel away from them.

We can better illustrate how the message travels up and down the structure by representing the tree with root 9 (see Figure 1(b)) as the associated tree structure where the message travels only downwards.

We will consider a generic node j surrounded by neighbours p, i, k; one of these may be a parent and the others children, or all of them children. The message, denoted by O, might come from any of them and travel then to the others. The message will come with other symbols that help the system implementing the algorithm. We will conceive various rules allowing the message received from one of its neighbours to travel through j towards its other neighbours. We will consider four distinct cases illustrated by different types of P systems.

Case 1. Initially j consists of an empty multiset of objects and the rules are

- (i) $p' \to ik$, $i' \to pk$, $k' \to pi$;
- (ii) $p \to (j'O)_p$, $i \to (j'O)_i$, $k \to (j'O)_k$.

When a message comes from a neighbour, p for instance, then the corresponding multiset, Op' in this case, is received. In the first step p' is transformed into ik by using a rule of type (i). Next these two symbols trigger rules from (ii) which in turn send the multiset Oj' to the neighbours i and k, respectively.

Case 2. Like in the previous case, j consists of an initial empty multiset; the rules are

$$p \to (jO)_i(jO)_k,\ i \to (jO)_p(jO)_k,\ k \to (jO)_p(jO)_i.$$

In this case when p is received together with O it will send jO to i and to k by using the first rule. We notice that the message is processed and passed on to its neighbours in one single step.

Case 3. The compartment j contains the multiset pik and the rules

$$pc \to (j'Oc^{n_{j,p}})_p|\neg p',\ ic \to (j'Oc^{n_{j,i}})_i|\neg i',\ kc \to (j'Oc^{n_{j,k}})_k|\neg k'.$$

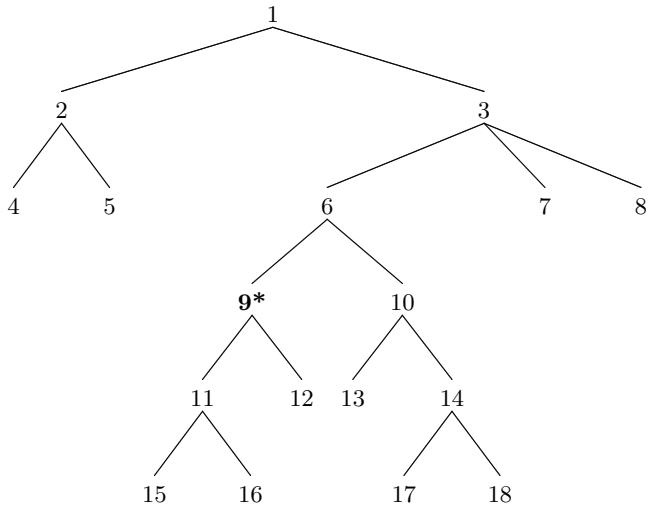
In this case j receives from p the multiset $p'Occ$. The symbol p' acts as an inhibitor of the first rule, preventing it to resend O back to p. The two c's allow the second and third rules to be executed. In the above rules $n_{j,h}$ defines the number of neighbours of h, excluding j, $h \in \{p, i, k\}$. These rules are applied in one step.

Case 4. The region j contains the multiset pik and the rules

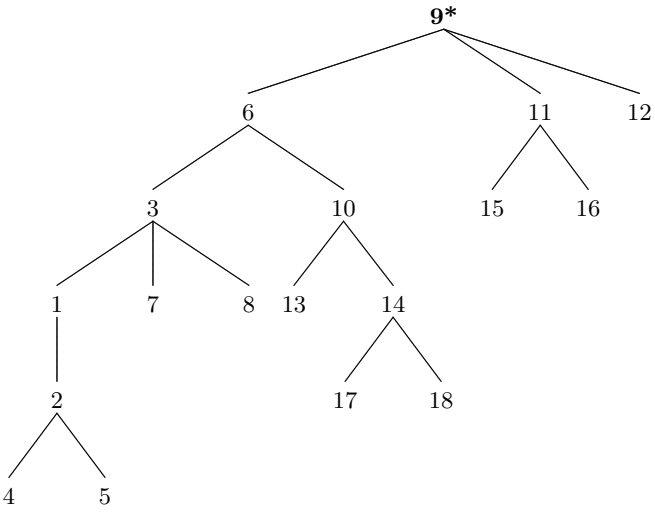
- (i) $c \to x^2$,
- (ii) $px \to (j'Oc)_p|\neg p'$, $ix \to (j'Oc)_i|\neg i'$, $kx \to (j'Oc)_k|\neg k'$.

Once j receives from its neighbour p the intended message through the multiset $p'Oc$, the rule (i) is executed and two x's are produced; then they will allow the second and third rule from the set (ii) to send appropriate messages to neighbours i and k.

These four cases have a constant time complexity, either one or two steps. We now analyse the correlations between the format of rules in a compartment and the



(a) Tree describing a membrane structure; the start node for broadcasting is 9



(b) Broadcasting information from node 9

Fig. 1. Trees illustrating the membrane structure of a P system and the broadcasting principle

number of its neighbours. More precisely, if we refer to the region j then for each neighbour the following happens: all the rules are affected in the first two cases; only two rules are changes in the third case and only three in the last one. It follows that the last two cases have a lower complexity than the first two with respect to the execution steps and number of changes made. We will consider the third case in our further investigations. This case, although very attractive due to its low complexity, with respect to number of steps, and relative robustness to changes, requires to assess in advance the number of neighbours for each compartment. We will consider this case for the example described in Figure 1(a).

Example 1. Let us consider a more general situation whereby a membrane j is included in p and contains k membranes $i_1, \ldots, i_k$. The region j consists of a multiset composed of the identifiers of the outer membrane, p, and inner membranes $i_1, \ldots, i_k$, i.e., its **close neighbours**. Formally this is given by $M_j = \{p, i_1, \cdots, i_k\}$. We will adopt this notation for multisets, instead of string based, due to numbers used as symbols in the notation below. Given the membrane structure defined by the tree in Figure 1(a), the membrane 9 is part of membrane 6 and contains 11 and 12. The membrane structure is provided by

$$\mu = [[[]_4[]_5]_2[[[[[]_{15}[]_{16}]_{11}[]_{12}]_{\mathbf{9}}[[]_{13}[[]_{17}[]_{18}]_{14}]_{10}]_6[]_7[]_8]_3]_1$$

the initial multisets are:

$$\begin{array}{llll} M_1 = \{2,3\} & M_2 = \{1,4,5\} & M_3 = \{1,6,7,8\} & M_4 = \{2\} \\ M_5 = \{2\} & M_6 = \{3,9,10\} & M_7 = \{3\} & M_8 = \{3\} \\ M_9 = \{6,11,12\} & M_{10} = \{6,13,14\} & M_{11} = \{9,15,16\} & M_{12} = \{9\} \\ M_{13} = \{10\} & M_{14} = \{10,17,18\} & M_{15} = \{11\} & M_{16} = \{11\} \\ M_{17} = \{14\} & M_{18} = \{14\} & & \end{array}$$

The rules of j are:

$$pc \rightarrow (j'O)_p(c_p)^{n_{j,p}} | \neg p';$$
$$i_s c \rightarrow (j'O)_{i_s}(c_{i_s})^{n_{j,i_s}} | \neg i'_s, \quad s = 1, \cdots, k;$$

where:

- like in Case 3 presented above, p', i'_s are inhibitors (a rule above is applied when there is no p' or i'_s, respectively, in membrane j), O is the message that will be sent, c is an object which is associated with a communication between two membranes;
- $n_{j,p}$, n_{j,i_s} are integer values defining the number of non-visited neighbours of p, i_s, respectively; it is easy to work out the relationship between the format of a rule and the number of non-visited descendants of the neighbour associated with the rule.

We briefly describe the first two steps of the broadcasting algorithm in this case.

Step 1. In the membrane that initiates the broadcasting are injected an object O and a number of objects c, one for every neighbour.

For example, if the starting membrane is $j = 9$, like in Figure 1(a), then we have the initial multiset M_9 and the additional symbols mentioned above leading to the multiset $\{6, 11, 12, O, c, c, c\}$; the rules are

$$\begin{aligned} R_9 = \quad & \{r_{9,6} : 6c \to (9'O)_6(c_6)^{n_{9,6}} | \neg 6', \\ & r_{9,11} : 11c \to (9'O)_{11}(c_{11})^{n_{9,11}} | \neg 11', \\ & r_{9,12} : 12c \to (9'O)_{12}(c_{12})^{n_{9,12}} | \neg 12'\} \end{aligned}$$

After these rules are applied in membrane 9, the objects $6, 11, 12, c, c, c$ are consumed and only an O remains in this membrane showing that the message has been received.

Step 2. Since this step onwards it is easy to follow the route of messages travelling through the system by representing it as a tree with root 9 as in Figure 1(b). If in Step 1 we consider $n_{9,6} = 2$, $n_{9,11} = 2$ and $n_{9,12} = 0$, then in the membranes $6, 11, 12$ which are neighbours of 9, the multisets will be: $\{3, 9, 10, 9', O, c, c\}$, $\{9, 15, 16, 9', O, c, c\}$, $\{9, 9', O\}$, respectively; the rules will be:

$$\begin{aligned} R_6 = \quad & \{r_{6,3} : 3c \to (6'O)_3(c_3)^{n_{6,3}} | \neg 3', \\ & r_{6,9} : 9c \to (6'O)_9(c_9)^{n_{6,9}} | \neg 9', \\ & r_{6,10} : 10c \to (6'O)_{10}(c_{10})^{n_{6,10}} | \neg 10'\} \end{aligned}$$

$$\begin{aligned} R_{11} = \quad & \{r_{11,9} : 9c \to (11'O)_9(c_9)^{n_{11,9}} | \neg 9', \\ & r_{11,15} : 15c \to (11'O)_{15}(c_{15})^{n_{11,15}} | \neg 15', \\ & r_{11,16} : 16c \to (11'O)_{16}(c_{16})^{n_{11,16}} | \neg 16'\} \end{aligned}$$

$$R_{12} = \{r_{12,9} : 9c \to (12'O)_9(c_9)^{n_{12,9}} | \neg 9'\}$$

The rules $r_{6,3}$, $r_{6,10}$, $r_{11,15}$, $r_{11,16}$ are applied and the following multisets are obtained $\{O\}$, $\{9, 9', O\}$, $\{9, 9', O\}$, $\{9, 9', O\}$, in regions $9, 6, 11, 12$, respectively.

If in Step 1 we consider $n_{9,6} = 0$ or $n_{9,6} = 1$, then at least one of the rules $r_{6,3}$ or $r_{6,10}$ cannot be applied as a c is missing and then in the corresponding hierarchy of compartments the message O is not received. [4] The multiset associated with region 6 becomes $\{3, 9, 9', O\}$, where 3 is the non-visited compartment together with its neighbours.

If in Step 1 it is considered $n_{9,6} > 2$ then the multiset is $\{3, 9, 10, 9', O, c^{n_{9,6}}\}$, and by applying the two existing rules, it becomes $\{9, 9', O, c^{n_{9,6}-2}\}$.

The process restarts from the compartments that have been affected by the communication rules in Step 2.

From this example we observe the following regarding the values $n_{j,i}$ involved.

[4] $r_{6,9}$ can not be applied due to the inhibitor $9'$

- If the values $n_{j,i}$ are appropriately chosen then in each membrane we will eventually get an O and no c.
- If $n_{j,i}$ is less than the expected value then for at least one hierarchy of compartments the message O does not travel to it.
- If $n_{j,i}$ has a bigger value then in some compartments we will have some more c's.
- Some $n_{j,i}$ do not count, i.e., those where the inhibitors i' are present. For instance: $n_{6,9}$, $n_{11,9}$, $n_{15,11}$ etc.
- For the membrane structure given in Figure 1(a), the solution is: $n_{9,6} = 2$, $n_{9,11} = 2$, $n_{9,12} = 0$, $n_{6,3} = 3$, $n_{6,10} = 2$, $n_{11,15} = 0$, $n_{11,16} = 0$, $n_{3,1} = 1$, $n_{3,7} = 0$, $n_{3,8} = 0$, $n_{10,13} = 0$, $n_{10,14} = 2$, $n_{1,2} = 2$, $n_{14,17} = 0$, $n_{14,18} = 0$, $n_{2,4} = 0$, $n_{2,5} = 0$; the other $n_{i,j}$ do not count.
- The number of $n_{i,j}$ values that are relevant is the same as the number of pairs parent-child in the membrane structure and is equal to the number of compartments - 1.
- By using the above values $n_{i,j}$, the P system will end up with the multisets below, where M_j is this multiset for the compartment j:

$$\begin{array}{lll} M_1 = \{3, 3', O\} & M_2 = \{1, 1', O\} & M_3 = \{6, 6', O\} \\ M_4 = \{2, 2', O\} & M_5 = \{2, 2', O\} & M_6 = \{9, 9', O\} \\ M_7 = \{3, 3', O\} & M_8 = \{3, 3', O\} & M_9 = \{O\} \\ M_{10} = \{6, 6', O\} & M_{11} = \{9, 9', O\} & M_{12} = \{9, 9', O\} \\ M_{13} = \{10, 10', O\} & M_{14} = \{10, 10', O\} & M_{15} = \{11, 11', O\} \\ M_{16} = \{11, 11', O\} & M_{17} = \{14, 14', O\} & M_{18} = \{14, 14', O\} \end{array}$$

- Given the non-determinism of the P system, for the same values of some parameters we can have different number of messages sent. For instance if $n_{9,6} = 1$, then $M_6 = \{3, 9, 10, 9', O, c\}$. If $r_{6,3}$ is applied then the hierarchy of compartments starting with 10 remains without messages (5 compartments without O). Similarly, if $r_{6,10}$ is applied then the 7 compartment occurring in the subtree rooted in 3 remained non-visited – see Figure 1(b).

4 Tuning the P system

In order to tune the system the values $n_{i,j}$ have to be identified. In the following a further transformation of the system is provided together with a more abstract representation.

The X-machine associated to the P system. According to the broadcasting problem defined above the values $n_{i,j}$ have to be found and we will apply an evolutionary approach using genetic algorithms to find these values. In order to apply it we will transform the cell-like structure of the system into a tree based structure. For a membrane structure μ we will consider as tree root the node from which the broadcast starts. For the P system presented in Example 1, node 9 will be the tree root - see Figure 1(b). We can further abstract the problem and define

each communication between two nodes i, j as a function $f_{i,j}$ with $n_{i,j}$ as its parameter describing the number of non-visited neighbours. It is easy to observe that the functions emerging from the same node will be executed in parallel, maybe together with other functions emerging from other nodes, they are independent of each other and an interleaving strategy can be adopted. In this case sequences of functions can be considered. A state machine or an X-machine can be defined by considering all possible interleavings of the arcs coming out of the nodes of a subtree. In the case presented in Example 1 the initial node is 9 and we distinguish three cases; when a c will be in 9 then we have three non-deterministic choices from 9 to each of the neighbours, the arcs being $f_{9,x}$ where $x \in \{6, 11, 12\}$; when two c's are in 9 then there are 6 non-deterministic choices: for each state defined by a pair $\{x, y\}$, $x, y \in \{6, 11, 12\}$, $x \neq y$, two non-deterministic sequences $f_{9,x}, f_{9,y}$ and $f_{9,y}, f_{9,x}$ can be conceived; for three or more c's there are again six non-deterministic choices from 9 to the state $\{6, 11, 12\}$, given by all the possible combinations of sequences of three functions $f_{9,x}$, $x \in \{6, 11, 12\}$. From each of the above seven states, $\{6\}, \{11\}, \{12\}, \{6, 11\}$ $\{6, 12\}, \{11, 12\}, \{6, 11, 12\}$ the construction of the machine follows the following steps: the arcs of the subtrees of roots specified by these states are shuffled. All shuffled routes starting in a given state are equivalent as the order of executing these functions does not matter.

5 Experiments and results

The experiments performed aimed to determine the unknown elements of a P system, more precisely the values $n_{i,j}$, using genetic algorithms. Considering that the structure of the P system contains m compartments, the number of parameter values that should be discovered is $m - 1$. In order to determine these values $n_{i,j}$, only the tree structure of the P system was used. Each candidate solution was encoded by an integer vector with $m - 1$ components, ranging from 0 to 10 and, consequently, the search space size was 11^{m-1}. The JGAP package (Java Genetic Algorithms and Genetic Programming Package) [9] was used for an elitist genetic algorithm implementation. The crossover operator has a great impact on the success of the genetic algorithm and the one chosen for this problem was the uniform crossover [7] (it is not part of the current JGAP version, but the package can be quickly extended with others operators). For selection we used a *BestChromosomesSelector* with the rate 0.8, which takes the top 80% individuals into the next generation, according to their fitness. The mutation operator employed had a 1/12 mutation rate.

The experiments performed considered trees having different number of nodes: 10, 15, 20, 25, 30, 35, 40, 45, 50. Obviously, it is more difficult to find a solution for a tree with 50 nodes (49 unknown variables) than for a tree with 10 nodes (and 9 unknown parameters). Due to the fact that the tree structure might have (or not) an influence on the problem considered, the following types of trees were considered:

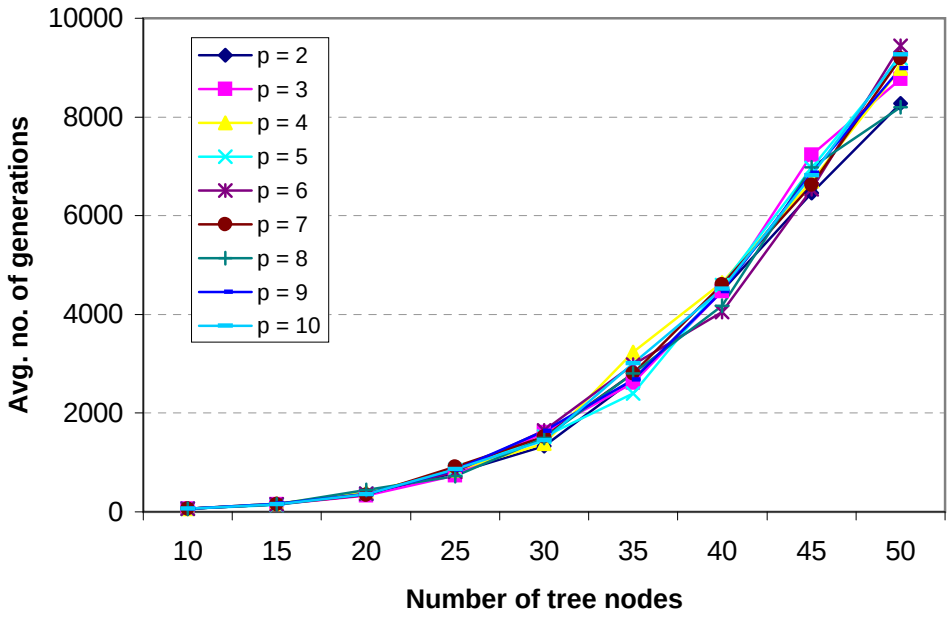


Fig. 2. Average number of generations for trees with fixed number of sons p

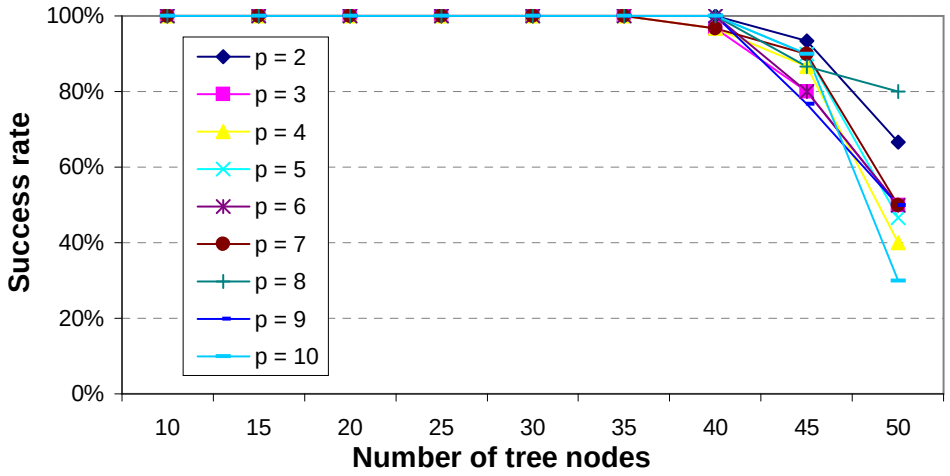


Fig. 3. Success rate for trees with a fixed number of sons p

1. *Trees with fixed number of sons:* each node has exactly p sons, excepting the leafs and eventually the last non-leaf node. For example, if the tree has $m = 10$ nodes and we consider $p = 3$, the root and its direct descendants will have exactly three sons. If $m = 10$ and $p = 4$, then the tree will have 4 direct descendants from the root, 4 for another node and only 1 descendant for another node.
2. *Trees with a random number of sons:* each non-leaf node can have a different number of sons, randomly chosen, with an equal probability, from the set {1,. . . ,p}.

In both cases, for each number of nodes $m \in \{10, 15, 20, 25, 30, 35, 40, 45, 50\}$ (corresponding to compartments in the P system) we considered all the values $p \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$. A tree was generated according to the structural criterion 1 or 2 and the unknown parameters values $n_{i,j}$ were searched using a genetic algorithm. The fitness function simulated a broadcasting (transmission) in the tree, starting from the root and using the parameters $n_{i,j}$. At the end of the transmission, each candidate solution was evaluated by counting the unvisited nodes and the extra messages sent to the nodes. For this we used the formula

$$fitness = \lambda \cdot no\_of\_unvisited\_nodes + no\_of\_extra\_messages,$$

where:

- $no\_of\_unvisited\_nodes$ represents the number of nodes where the message was not received: at the end of the computation, the membrane (node) does not contain any object O;
- $no\_of\_extra\_messages$ represents the number of extra objects c, present in the nodes at the end of the computation, that cannot be consumed;

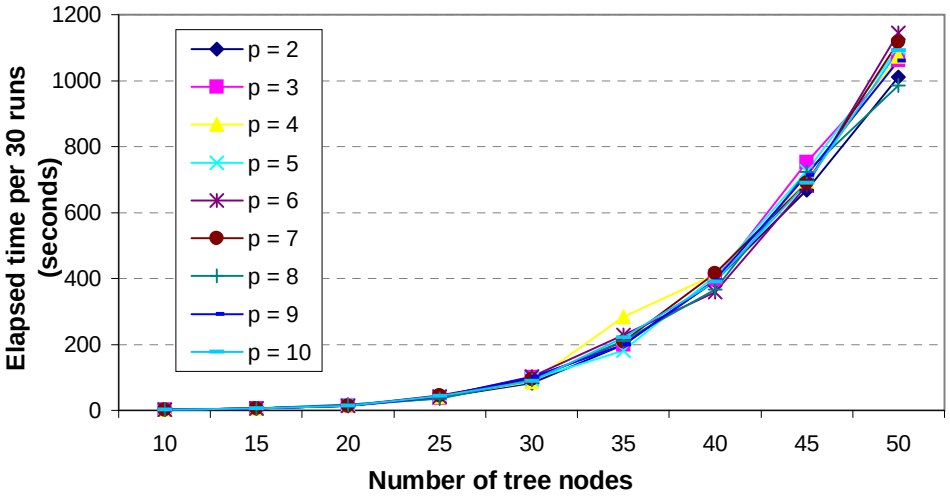


Fig. 4. Elapsed time for trees with a fixed number of sons p

- λ is a positive penalty (or weight) parameter which gives more importance to the $no\_of\_unvisited\_nodes$ or to the $no\_of\_extra\_messages$

Experimentally, we noticed that a function for which $\lambda > 1$ guided better the search than in the case in which $\lambda = 1$. After checking the convergence of the genetic algorithm on a few test trees, we decided to further use $\lambda = 10$, this way giving a higher penalty to the values $n_{i,j}$ which leave more unvisited nodes. The following termination criteria for the genetic algorithm were used: A) $fitness = 0$ (the solution was found: all the tree nodes were visited, with no extra messages sent) and B) the maximum allowed number of generations (10000) was reached. The population size used in these experiments was in all cases of 20 individuals.

For each combination, given by the structural criterion 1 or 2, the number of nodes in the tree $m \in \{10, 15, 20, \ldots, 50\}$ and the number of sons for each node $p \in \{2, 3, \ldots, 10\}$ the genetic algorithm was run 30 times. After each run, the best solution obtained, its fitness and the current generation were retained. The Tables 1,2,3,4 present, for each set of 30 the runs the following information: m = number of nodes in the tree, p = number of sons, the search space dimension for each case and the success rate for the 30 runs. Also, the mean and the standard deviation are shown for the best fitness function values (MF, SF) and for the number of generations (MG, SG), after 30 runs. The last column from the table shows the cumulated duration of the 30 runs, expressed in seconds.

We will refer only to results obtained for trees with fixed number of sons as for trees with random number of descendants the results are very similar. The average number of generations (Figure 2) and the time elapsed to get the solution (Figure 4) grow proportional to the number of nodes in the tree. The maximum allowed number of generations for the GA was set to 10000. Consequently, the success rates were very high for trees with less than 45 nodes (for which the solution was found in less generations) and then almost halves for trees with 50 nodes (Figure 3).

6 Conclusions

In this paper a method to determine the rules of a P system that models the broadcasting algorithm is introduced. Naturally, the number of unknown parameter values $n_{i,j}$ increases with the compartments number and consequently the search space size grows also. The search space size is obviously c^{no_par}, where c is the number of possible values for one parameter $n_{i,j}$ and no_par is the number of unknown parameters. The average number of generations and the elapsed time needed to find a solution increase when the search space is very large. If the maximum allowed number of generations is not high enough, the GA might end unsuccessful. One possible solution to overcome this is to increase the maximum allowed number of generations for the GA. Others solutions can be: using hybrid approaches, i.e. combining GAs with local search techniques (like hill climbing) and

developing new GAs operators, suited for this problem (the crossover operator has in particular a great impact on the GA).

The method is described in a more general context of an abstract X-machine that captures some specific aspects of the P system, namely the size of the rules. Given that similar approaches to map P systems into X-machines prove to be very effective in testing these systems [8], we can conclude that such testing strategies developed for associated X-machines can be applied in the case of the broadcasting problem as well. Hence, we can provide a powerful method to estimate the P system that models the broadcasting problem and then test the implementation based on this model.

Further studies will aim to improve the precision and efficiency of the method discussed in this paper and to extend it to other classes of P systems.

Acknowledgements.

The research of RL, FI and MG is supported by CNCSIS grant IDEI no.496/2009, *An integrated evolutionary approach to formal modelling and testing* (EvoMT). The research of GZ is supported by the National Natural Science Foundation of China (60702026), the Scientific and Technological Funds for Young Scientists of Sichuan and the Open Foundation of Engineering Research Centre of Safety Transportation of the Ministry of Education of China. The authors would like to thank all the referees for their helpful comments.

References

1. Besozzi, D., Zandron, C., Mauri, G., Sabadini, N.: P systems with gemmation of mobile membranes. In: Lecture Notes in Computer Science. Volume 2202, London, UK, Springer-Verlag (2001) 136–153
2. Bottoni, P., Martín-Vide, C., Păun, G., Rozenberg, G.: Membrane systems with promoters/inhibitors. Acta Informatica **38**(10) (2002) 695–720
3. Castellini, A., Manca, V.: Learning regulation functions of metabolic systems by artificial neural networks. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2009), ACM Publisher (2009), to appear
4. Castellini, A., Manca, V., Suzuki, Y.: Metabolic P system flux regulations by artificial neural networks. In: Proceedings of the Tenth Workshop on Membrane Computing (WMC10). (2009), to appear
5. Cavaliere, M., Mardare, R.: Partial knowledge in membrane systems: A logical approach. In: Proceedings of the WMC7 (2006) 242–260 and Lecture Notes in Computer Science. Volume 4361. Membrane Computing, WMC2006, Leiden, Revised, Selected and Invited Papers, Hoogeboom, H.J., Păun, Gh., Rozenberg, G., Salomaa, A., eds., Springer (2006), 279–297
6. Ciobanu, G.: Distributed algorithms over communicating membrane systems. Biosystems **70**(2) (2003) 123–133
7. Drake, S.: Uniform crossover revisited: Maximum disruption in real-coded gas. In: GECCO. (2003) 1576–1577

8. Ipate, F., Gheorghe, M.: Testing non-deterministic stream X-machine models and P systems. Electr. Notes Theor. Comput. Sci. **227** (2009) 113–126
9. K. Meffert et al.: JGAP - Java Genetic Algorithms and Genetic Programming Package
10. Păun, Gh.: Computing with membranes. Journal of Computer and System Sciences **61**(1) (2000) 108–143
11. Păun, Gh.: Membrane computing. An introduction. Springer, Berlin (2002)
12. Păun, Gh., Rozenberg, G.: A guide to membrane computing. Theoretical Computer Science **287**(1) (2002) 73–100
13. Romero-Campero, F.J., Cao, H., Camara, M., Krasnogor, N.: Structure and parameter estimation for cell systems biology models. In: GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation, ACM (2008) 331–338

m	p	Space size	Succ.	MF	SF	MG	SG	Dur.
10	2	2.36E+09	100.0 %	0.00	0.00	65.50	29.60	1
10	3	2.36E+09	100.0 %	0.00	0.00	62.77	25.38	1
10	4	2.36E+09	100.0 %	0.00	0.00	66.13	33.93	1
10	5	2.36E+09	100.0 %	0.00	0.00	63.13	21.65	1
10	6	2.36E+09	100.0 %	0.00	0.00	68.43	28.07	1
10	7	2.36E+09	100.0 %	0.00	0.00	59.07	22.45	1
10	8	2.36E+09	100.0 %	0.00	0.00	63.77	26.55	1
10	9	2.36E+09	100.0 %	0.00	0.00	63.50	20.34	1
10	10	2.36E+09	100.0 %	0.00	0.00	67.10	29.93	1
15	2	3.80E+14	100.0 %	0.00	0.00	161.07	76.80	5
15	3	3.80E+14	100.0 %	0.00	0.00	153.53	63.31	5
15	4	3.80E+14	100.0 %	0.00	0.00	161.53	61.55	5
15	5	3.80E+14	100.0 %	0.00	0.00	140.13	51.84	5
15	6	3.80E+14	100.0 %	0.00	0.00	154.70	76.02	5
15	7	3.80E+14	100.0 %	0.00	0.00	154.03	71.19	5
15	8	3.80E+14	100.0 %	0.00	0.00	156.83	43.10	5
15	9	3.80E+14	100.0 %	0.00	0.00	168.27	69.02	5
15	10	3.80E+14	100.0 %	0.00	0.00	153.80	63.36	5
20	2	6.12E+19	100.0 %	0.00	0.00	363.10	188.67	14
20	3	6.12E+19	100.0 %	0.00	0.00	327.23	124.68	13
20	4	6.12E+19	100.0 %	0.00	0.00	388.43	158.48	15
20	5	6.12E+19	100.0 %	0.00	0.00	370.63	213.60	15
20	6	6.12E+19	100.0 %	0.00	0.00	364.13	135.54	14
20	7	6.12E+19	100.0 %	0.00	0.00	348.33	152.45	14
20	8	6.12E+19	100.0 %	0.00	0.00	442.20	178.49	17
20	9	6.12E+19	100.0 %	0.00	0.00	354.73	154.01	14
20	10	6.12E+19	100.0 %	0.00	0.00	357.30	142.46	14
25	2	9.85E+24	100.0 %	0.00	0.00	782.77	250.66	39
25	3	9.85E+24	100.0 %	0.00	0.00	730.87	228.65	37
25	4	9.85E+24	100.0 %	0.00	0.00	834.83	316.04	42
25	5	9.85E+24	100.0 %	0.00	0.00	855.97	358.82	43
25	6	9.85E+24	100.0 %	0.00	0.00	808.17	369.63	41
25	7	9.85E+24	100.0 %	0.00	0.00	915.60	306.14	46
25	8	9.85E+24	100.0 %	0.00	0.00	721.67	327.35	36
25	9	9.85E+24	100.0 %	0.00	0.00	847.90	336.24	42
25	10	9.85E+24	100.0 %	0.00	0.00	859.47	313.92	43
30	2	1.59E+30	100.0 %	0.00	0.00	1329.13	488.55	83
30	3	1.59E+30	100.0 %	0.00	0.00	1616.13	850.51	100
30	4	1.59E+30	100.0 %	0.00	0.00	1377.40	516.99	86
30	5	1.59E+30	100.0 %	0.00	0.00	1522.27	642.85	94

Table 1. Statistics for trees with m nodes and fixed number of sons p

m	p	Space	Succ.	MF	SF	MG	SG	Dur.
30	6	1.59E+30	100.0 %	0.00	0.00	1653.27	549.80	102
30	7	1.59E+30	100.0 %	0.00	0.00	1523.37	585.38	94
30	8	1.59E+30	100.0 %	0.00	0.00	1493.30	512.72	92
30	9	1.59E+30	100.0 %	0.00	0.00	1643.13	659.68	101
30	10	1.59E+30	100.0 %	0.00	0.00	1452.80	490.25	89
35	2	2.55E+35	100.0 %	0.00	0.00	2678.50	972.70	200
35	3	2.55E+35	100.0 %	0.00	0.00	2611.27	932.03	199
35	4	2.55E+35	100.0 %	0.00	0.00	3237.53	1537.68	284
35	5	2.55E+35	100.0 %	0.00	0.00	2398.23	609.77	181
35	6	2.55E+35	100.0 %	0.00	0.00	2984.70	844.46	228
35	7	2.55E+35	100.0 %	0.00	0.00	2808.17	869.71	209
35	8	2.55E+35	100.0 %	0.00	0.00	2810.83	929.17	211
35	9	2.55E+35	100.0 %	0.00	0.00	2673.77	857.52	199
35	10	2.55E+35	100.0 %	0.00	0.00	3004.70	1077.03	220
40	2	4.11E+40	100.0 %	0.00	0.00	4476.57	1587.89	392
40	3	4.11E+40	96.7 %	0.03	0.18	4464.57	1544.74	397
40	4	4.11E+40	96.7 %	0.03	0.18	4643.00	1670.95	412
40	5	4.11E+40	100.0 %	0.00	0.00	4592.83	1640.95	407
40	6	4.11E+40	100.0 %	0.00	0.00	4046.40	1184.09	358
40	7	4.11E+40	96.7 %	0.03	0.18	4607.63	1851.40	415
40	8	4.11E+40	100.0 %	0.00	0.00	4175.73	1212.75	366
40	9	4.11E+40	100.0 %	0.00	0.00	4471.87	1566.23	394
40	10	4.11E+40	100.0 %	0.00	0.00	4513.00	1465.79	390
45	2	6.63E+45	93.3 %	0.07	0.25	6464.13	2055.62	667
45	3	6.63E+45	80.0 %	0.27	0.58	7239.77	1869.96	753
45	4	6.63E+45	86.7 %	0.17	0.46	6700.20	2103.96	695
45	5	6.63E+45	90.0 %	0.10	0.31	6964.23	2095.90	725
45	6	6.63E+45	80.0 %	0.27	0.58	6541.43	2342.57	682
45	7	6.63E+45	90.0 %	0.13	0.43	6620.33	1898.96	690
45	8	6.63E+45	86.7 %	0.17	0.46	6975.67	1719.86	723
45	9	6.63E+45	76.7 %	0.27	0.52	6864.73	2304.19	714
45	10	6.63E+45	90.0 %	0.10	0.31	6811.03	1914.10	689
50	2	1.07E+51	66.7 %	0.47	0.78	8275.27	1624.30	1011
50	3	1.07E+51	50.0 %	0.80	0.89	8766.20	1434.48	1063
50	4	1.07E+51	40.0 %	0.80	0.89	8981.07	1523.70	1073
50	5	1.07E+51	46.7 %	0.67	0.76	9189.17	1309.14	1100
50	6	1.07E+51	50.0 %	0.57	0.63	9443.93	991.21	1145
50	7	1.07E+51	50.0 %	0.70	0.84	9189.77	1131.65	1117
50	8	1.07E+51	80.0 %	0.33	0.76	8198.33	1419.55	985
50	9	1.07E+51	50.0 %	0.73	0.87	8985.03	1310.33	1061
50	10	1.07E+51	30.0 %	1.03	0.89	9262.57	1306.81	1092

Table 2. Statistics for trees with m nodes and fixed number of sons p

m	p	Space	Succ.	MF	SF	MG	SG	Dur.
10	2	2.36E+09	100.0 %	0.00	0.00	63.27	32.43	1
10	3	2.36E+09	100.0 %	0.00	0.00	58.27	23.10	1
10	4	2.36E+09	100.0 %	0.00	0.00	58.80	17.81	1
10	5	2.36E+09	100.0 %	0.00	0.00	72.73	25.61	1
10	6	2.36E+09	100.0 %	0.00	0.00	66.50	24.56	1
10	7	2.36E+09	100.0 %	0.00	0.00	64.47	29.34	1
10	8	2.36E+09	100.0 %	0.00	0.00	57.10	20.66	1
10	9	2.36E+09	100.0 %	0.00	0.00	57.13	17.19	1
10	10	2.36E+09	100.0 %	0.00	0.00	66.30	25.95	1
15	2	3.80E+14	100.0 %	0.00	0.00	155.43	67.94	5
15	3	3.80E+14	100.0 %	0.00	0.00	148.43	49.25	5
15	4	3.80E+14	100.0 %	0.00	0.00	158.93	73.09	5
15	5	3.80E+14	100.0 %	0.00	0.00	163.67	62.21	5
15	6	3.80E+14	100.0 %	0.00	0.00	159.40	60.32	5
15	7	3.80E+14	100.0 %	0.00	0.00	152.60	55.89	5
15	8	3.80E+14	100.0 %	0.00	0.00	147.73	49.97	5
15	9	3.80E+14	100.0 %	0.00	0.00	152.03	63.03	5
15	10	3.80E+14	100.0 %	0.00	0.00	153.90	58.05	5
20	2	6.12E+19	100.0 %	0.00	0.00	381.27	155.23	14
20	3	6.12E+19	100.0 %	0.00	0.00	379.93	149.30	15
20	4	6.12E+19	100.0 %	0.00	0.00	356.53	94.02	15
20	5	6.12E+19	100.0 %	0.00	0.00	358.53	161.98	14
20	6	6.12E+19	100.0 %	0.00	0.00	353.97	151.56	14
20	7	6.12E+19	100.0 %	0.00	0.00	328.90	110.80	13
20	8	6.12E+19	100.0 %	0.00	0.00	382.50	149.76	15
20	9	6.12E+19	100.0 %	0.00	0.00	407.43	179.24	16
20	10	6.12E+19	100.0 %	0.00	0.00	376.80	196.99	15
25	2	9.85E+24	100.0 %	0.00	0.00	882.93	475.38	44
25	3	9.85E+24	100.0 %	0.00	0.00	772.13	249.86	39
25	4	9.85E+24	100.0 %	0.00	0.00	823.43	364.62	41
25	5	9.85E+24	100.0 %	0.00	0.00	863.57	363.86	43
25	6	9.85E+24	100.0 %	0.00	0.00	833.20	495.31	42
25	7	9.85E+24	100.0 %	0.00	0.00	842.67	318.03	42
25	8	9.85E+24	100.0 %	0.00	0.00	834.87	291.64	42
25	9	9.85E+24	100.0 %	0.00	0.00	822.13	423.69	40
25	10	9.85E+24	100.0 %	0.00	0.00	834.23	323.54	42
30	2	1.59E+30	100.0 %	0.00	0.00	1549.30	680.43	94
30	3	1.59E+30	100.0 %	0.00	0.00	1519.50	622.27	93
30	4	1.59E+30	100.0 %	0.00	0.00	1785.00	589.93	109
30	5	1.59E+30	100.0 %	0.00	0.00	1475.80	761.75	94

Table 3. Statistics for trees with m nodes and variable number of sons between $\{1, \ldots, p\}$

m	p	Space	Succ.	MF	SF	MG	SG	Dur.
30	6	1.59E+30	100.0 %	0.00	0.00	1657.53	621.15	105
30	7	1.59E+30	100.0 %	0.00	0.00	1691.43	555.99	108
30	8	1.59E+30	100.0 %	0.00	0.00	1423.43	517.50	91
30	9	1.59E+30	100.0 %	0.00	0.00	1579.10	497.21	99
30	10	1.59E+30	100.0 %	0.00	0.00	1451.13	502.28	92
35	2	2.55E+35	100.0 %	0.00	0.00	2864.53	1163.11	218
35	3	2.55E+35	100.0 %	0.00	0.00	2700.27	832.91	205
35	4	2.55E+35	100.0 %	0.00	0.00	2822.23	1200.13	216
35	5	2.55E+35	100.0 %	0.00	0.00	3073.17	1217.87	236
35	6	2.55E+35	100.0 %	0.00	0.00	2491.67	781.10	192
35	7	2.55E+35	100.0 %	0.00	0.00	2426.27	785.50	186
35	8	2.55E+35	100.0 %	0.00	0.00	2681.93	859.48	203
35	9	2.55E+35	100.0 %	0.00	0.00	2952.83	1151.27	226
35	10	2.55E+35	100.0 %	0.00	0.00	2610.67	1133.39	193
40	2	4.11E+40	100.0 %	0.00	0.00	4112.87	1169.85	364
40	3	4.11E+40	100.0 %	0.00	0.00	4785.10	1656.32	420
40	4	4.11E+40	100.0 %	0.00	0.00	4557.37	1544.00	401
40	5	4.11E+40	100.0 %	0.00	0.00	4120.03	1417.94	363
40	6	4.11E+40	100.0 %	0.00	0.00	4534.57	1634.49	404
40	7	4.11E+40	100.0 %	0.00	0.00	4819.10	1268.60	422
40	8	4.11E+40	100.0 %	0.00	0.00	4281.47	1744.27	375
40	9	4.11E+40	96.7 %	0.03	0.18	4317.23	1719.93	378
40	10	4.11E+40	100.0 %	0.00	0.00	4343.13	1500.82	384
45	2	6.63E+45	83.3 %	0.17	0.38	6733.47	2096.95	690
45	3	6.63E+45	90.0 %	0.10	0.31	7226.00	1993.56	745
45	4	6.63E+45	83.3 %	0.17	0.38	6764.33	2299.93	693
45	5	6.63E+45	80.0 %	0.20	0.41	7079.87	1975.24	725
45	6	6.63E+45	93.3 %	0.07	0.25	6686.43	1886.45	686
45	7	6.63E+45	93.3 %	0.07	0.25	6328.57	1980.41	649
45	8	6.63E+45	86.7 %	0.13	0.35	6766.33	2225.37	690
45	9	6.63E+45	83.3 %	0.17	0.38	6997.10	1954.87	717
45	10	6.63E+45	93.3 %	0.07	0.25	6519.57	1973.89	671
50	2	1.07E+51	40.0%	0.80	0.89	9223.73	1262.43	1088
50	3	1.07E+51	40.0%	0.80	0.81	9554.10	699.02	1145
50	4	1.07E+51	33.3%	0.87	0.82	9311.43	1200.56	1097
50	5	1.07E+51	50.0%	0.67	0.76	8869.97	1615.60	1040
50	6	1.07E+51	63.3%	0.37	0.49	8775.83	1498.18	1034
50	7	1.07E+51	53.3%	0.60	0.77	8571.30	1755.25	1014
50	8	1.07E+51	56.7%	0.67	0.99	8782.73	1450.53	1051
50	9	1.07E+51	40.0%	0.73	0.69	8849.47	1684.96	1042
50	10	1.07E+51	50.0%	0.70	0.88	8994.07	1635.41	1063

Table 4. Statistics for trees with m nodes and variable number of sons between $\{1, \ldots, p\}$