POLYOMINOIDS AND UNIFORM ELECTION

ABDELAAZIZ EL HIBAOUI, NASSER SAHEB-DJAHROMI AND AKKA ZEMMARI*
LABRI, UNIVERSITÉ BORDEAUX I,
351 COURS DE LA LIBÉRATION
33405 TALENCE, FRANCE

Abstract. In this paper, a new structure for polyominoid graph is proposed. This structure is shown to be generated with some rules. A uniform probabilistic election algorithm in polyominoids is developed and studied. Indeed, the election process is considered as a distributed elimination algorithm in a polyominoid, which removes all active vertices one after the other, till there remains one single vertex: the leader.

The elimination algorithm is analyzed as a Markovian random process in continuous time. Our algorithm is totally fair in that all vertices have the same probability of being elected.
Key words: Distributed Algorithms, Election, Fairness.

1. Introduction

We consider distributed networks of processors [23]. They are presented as a connected graphs where vertices represent processors, and two vertices are connected by an edge if the corresponding processors have a direct communication link. The networks are asynchronous: processors cannot access a global clock and a message sent from a processor to a neighbor arrives within some finite but unpredictable time (asynchronous message passing). Labels are attached to vertices and sometimes to edges. The aim of an election problem is to choose exactly one element in the set of processors. Thus, starting from a configuration where all processors are in the same state, we must obtain a configuration where exactly one processor is in the state "leader" and all other processors are in the state "lost". The leader can be used subsequently to make decisions or to centralize some information. The election problem is well known and many solutions are available [1, 14, 15, 17, 20, 23]. It was first proposed by Le Lann [14].

The networks studied in this paper are anonymous and have a polyominoid topology. A polyominoid combines the tree and polyominos-structure (see Figure 1). Polyominoes have a long history, going back to the beginning of the 20^{th} century, but they were popularized in the present era by Golomb [11, 12] and Gardner [9, 10] in the *Scientific American columns*, "Mathematical Games". They were also studied by mathematicians [2, 3, 5, 6], because they constitute combinatoric objects having interesting properties. They have been the subject of intensive studies by physicists, thanks to their appropriateness for modeling several physical phenomena and are known under the name of animals in statistical physics (see [24] for more details). In computer science, their study has been motivated in different areas such as the VLSI circuit designs (see [16]) and image processing ([4]).

The main motivation behind this study is to introduce a uniform probabilistic distributed election algorithm over polyominoids. This algorithm is *totally fair*, i.e. it gives a same chance of being elected to all vertices of a polyominoid. The algorithm removes vertices of the polyominoid once their random lifetime delay

*e-mail: {hibaoui, saheb, zemmari}@labri.fr.

has been expired (the remaining graph should remain polyominoid). The analysis of the algorithm reveals the surprising fact that, wherever the vertex is placed in the polyominoid, it has the same probability of surviving as the others. The only investigation in this direction, known to the authors, is that of trees [19] .

Our distributed algorithm may be viewed as a randomized extension of a variant of [15], where random delays are introduced.

We consider cellular local computations which allow to modify the state (or label) of a vertex at each step. The new label depends on the previous one and those of its neighbors. The novelty of our approach is the use of random delays for relabeling. These delays are exponential random variables defined independently for active vertices. The parameter of the random variable for a vertex is equal to the attributed value assigned to the vertex. The process of relabeling continues until no more transformation is possible, i.e. a final configuration is reached. In this configuration, there is only one $L-$labeled vertex, considered as elected.

The paper is organized as follows. In Section 2, we introduce the preliminaries and basic notation. Polyominoids are introduced in this section as particular undirected graphs. We provide a set of rules generating the class of all polyominoids. The election algorithm is described in Section 3. The main result in Section 4 is the uniformity of the election on the set of vertices. Due to the space limitation we have to omit simple proofs. For more details, the reader is referred to [13].

2. Preliminaries and Notation

There are many definitions for polyominos and grid-like graphs in the literature, see [21, 18]. Traditionally, a polyomino is the set of cells situated in the interior of an orthogonal polygon drawn on a grid. We define polyominoids as finite graphs whose nodes are points from $\mathbb{Z}\times\mathbb{Z}$, where $\mathbb{Z}$ denotes the set of integers, possibly linked by the neighborhood relationship, defined in the sequel.

Throughout this paper, the vertices are points from $\mathcal{Z}=\mathbb{Z}\times\mathbb{Z}$. We use usual terms such as "up", "down", "right" and "left" on $\mathbb{Z}\times\mathbb{Z}$. The edges are links between pairs of points, i.e. sets of pairs of points of one of the forms $\{(x,y),(x+1,y)\}$ or $\{(x,y),(x,y+1)\}$, for some $x\in\mathbb{Z}, y\in\mathbb{Z}$. Two vertices $v=(x,y)$ and $v'=(x',y')$ of $\mathcal{Z}$ are *neighbors* if either $x=x'$ and $|y-y'|=1$ or else $y=y'$ and $|x-x'|=1$. We refer to each element of an edge e as its *end*. Let $\mathcal{T}$ be the set of all these edges and set $\mathbf{U}=(\mathcal{Z},\mathcal{T})$. A *cell* is a subgraph of $\mathbf{U}$, induced by a set $\{(x,y),(x+1,y),(x+1,y+1),(x,y+1)\}$ of four pairwise neighbor vertices. A *path* is a finite alternated sequence $\sigma=v_0,e_1,\ldots,e_k,v_k$ of $k+1$ vertices and k *different* edges ($k\geq 0$), such that each edge e_i has one end in v_{i-1} and the other one in v_i. We should note that a path may pass several times through a vertex but cannot borrow an edge more than once. The *length* of a path σ as above is k. For the sake of briefness in a path, we may drop edges, identifying σ by the sequence of vertices $v_0,v_1,\ldots,v_k$. If so, any pair of two successive terms v_i and v_{i+1} should constitute a unique set. A *cycle* is a path of length $k\geq 4$ for which the first vertex v_0 and the last one v_k coincide. $\mathbf{U}$ is bipartite i.e. all its cycles are of even length.

Given a cycle γ, one can easily define its inside vertices, see [22]. A vertex (x,y) is said to be *inside* a cycle $\gamma=(x_0,y_0),\ldots,(x_k,y_k)$, with $(x_0,y_0)=(x_k,y_k)$, if $card(\{i\mid y=y_i \text{ and } y\neq y_{i+1} \text{ and } x\leq x_i\})$ is odd (in the addition $i+1$ *modulo* k). According to this definition, the vertices of γ are inside γ.

A *polyominoid* is a partial subgraph $\mathbf{P}=(V,E)$ of $\mathbf{U}$ subject to the following conditions

(i) V is finite,
(ii) $\mathbf{P}$ is connected and

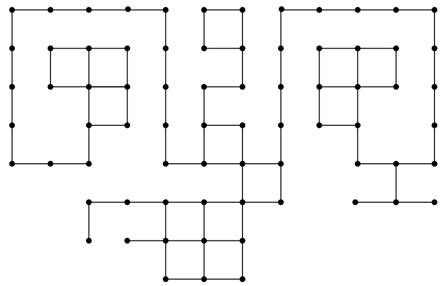

Figure 1. An example of polyominoid

(iii) $\mathbf{P}$ does not contain any hole, i.e. for all cycle γ in $\mathbf{P}$, the vertices inside γ are contained in V and if two neighbor vertices are inside γ, then the linking edge is in E.

It is easy to see that the last property is equivalent to the *tilability* of $\mathbf{P}$=(V, E), i.e. the set of vertices inside γ and their linking edges constitute a subgraph of a grid. The *size* of $\mathbf{P}$= (V, E) is the cardinal of V.

A polyominoid $\mathbf{Q}$= (V', E') is called a *subpolyominoid* of a polyominoid $\mathbf{P}$ = (V, E) if $V' \subseteq V$, $E' \subseteq E$ and $E' = E \cap \{\{u, v\} \mid u \in V', v \in V'\}$.

The class of polyominoids can be defined on $\mathbf{U}$ by induction in a distributive fashion as follows. The construction is totally distributive in that the application of rewriting rules requires only the knowledge of the neighboring areas in a bull of radius 2. Thus, the set of polyominoids can be generated by a context-free-like grammar. We define the set $\mathcal{P}$ of partial subgraphs of $\mathbf{U}$ by the following inductive rules:

(a) For any $(x, y) \in \mathcal{Z}$, $\mathbf{P}$=$(\{(x, y)\}, \emptyset)$ is in $\mathcal{P}$.
(b) Let $\mathbf{P}$=$(V, E) \in \mathcal{P}$. Consider two neighbor vertices v and v' such that $v \in V$ and $v' \notin V$. Then, $\mathbf{Q}$=$(V \cup \{v'\}, E \cup \{\{v, v'\}\})$ is in $\mathcal{P}$.
(c) Let $\mathbf{P}$=$(V, E) \in \mathcal{P}$. Suppose V contains 4 neighbor vertices $v_1 = (x, y), v_2 = (x+1, y), v_3 = (x+1, y+1), v_4 = (x, y+1)$, situated on a cell in $\mathbf{U}$, such that three edges of the cell on them are in E and the fourth one, say e, is not. Then, $\mathbf{Q}$=$(V, E \cup \{e\})$ is in $\mathcal{P}$.

At this stage, it is not obvious that $\mathcal{P}$ is the class of all polyominoids on $\mathbf{U}$. The following proposition shows the equivalence of the two definitions.

Proposition 1. *A partial subgraph* $\mathbf{P}$=(V, G) *of* $\mathbf{U}$ *is a polyominoid iff it belongs to* $\mathcal{P}$.

3. A Uniform Election Algorithm on Polyominoids

The asynchronous election algorithm, presented in this section, is designed for anonymous networks having a topology of polyominoid. Each vertex only knows the directions of the edges joining it to its neighbors, and knows neither the size of polyominoid nor its own coordinates in the plan. The solution in the general case consists in the computation of a spanning tree, and then election is started for every node. In our study, using the properties of polyominoids, we construct a distributed algorithm which chooses uniformly a vertex as the leader.

3.1. **The Distributed Election.** We now describe the algorithm through a graph relabeling system. Labels (or states) are attached to vertices. Our distributed algorithm is based on a rewriting system, introduced by Litovsky, Métivier and Zielonka [15].

We suppose that initially every vertex has the same label and we look for a noetherian graph rewriting system such that when, after some number of rewriting steps, we get an irreducible labeled graph the there is a special label that is attached to exactly one vertex; this vertex is considered as elected.

In this paper, we use a graph rewriting system enriched by random delays (a rule may be applied if its delay has expired). The graph rewriting system applied here uses forbidden contexts (a rule may be applied if it does not occur in a given forbidden context).

Let $\mathbf{P}= (V, E)$ be a polyominoid and let v a vertex of V. We introduce the set $\mathcal{L}$ of labels $\{N, A, B, L\}$ where N encodes the neutral state, A encodes the active state, B encodes the lost state and lastly L encodes the elected state. Initially, every vertex is of weight $w = 1$ and is $N-$labeled.

Given a polyominoid $\mathbf{P}= (V, E)$, the algorithm works on $\mathbf{P}$ as follows. Any $N-$labeled vertex v decides locally if it is active or not, according to the following rules :

R_0 : If the degree of vertex v is null ($deg(v) = 0$), then the single vertex which constitutes the polyominoid is the elected vertex. This vertex is considered as an *active* vertex.

R_1 : If the degree of vertex v is 1 ($deg(v) = 1$), then the vertex v becomes *active* and it generates its lifetime delay which is an exponentially r.v. (random variable) having its weight as the parameter. Whenever its lifetime has expired, it is removed with its unique incident edge. At this time, the vertex adjacent to the removed one in $\mathbf{P}$, collects the weight of this removed vertex, adding it to its weight.

R_2 : If $deg(v) = 2$, then whenever v is a upper-left most vertex or lower-left most of cell, then it becomes *active* and when its lifetime has expired, it is removed with its incident edges and its right neighbor recuperates its weight.

More precisely, let $\{(x, y), (x + 1, y), (x, y + 1), (x + 1, y + 1)\}$ be a cell, if the degree of (x, y) is 2 then (x, y) is *active* and once its lifetime has expired its neighbor $(x + 1, y)$ picks up its weight. In the same way and by the horizontal symmetry, if $deg((x, y + 1)) = 2$ then $(x, y + 1)$ is *active* and its neighbor $(x + 1, y + 1)$ collects its weight.

R_3 : If $deg(v) = 3$, then if v belongs to two cells and no edge on its left side is found, i.e. v has only one horizontal edge, then v becomes *active* and when its lifetime has expired then its right neighbor recuperates its weight.

So, let $\{(x, y), (x+1, y), (x, y+1), (x+1, y+1)\}$ and $\{(x, y), (x+1, y), (x, y-1), (x-1, y-1)\}$ two cells, if degree (x, y) is 3 then (x, y) becomes *active* and its neighbor $(x + 1, y)$ recuperates its weight once its lifetime has expired.

In the sequel, we need the following definition. A vertex which belongs to a maximal cycle in $\mathbf{P}$ is called *peripheral vertex.*

Lemma 1. *Let v be an active vertex of degree 2 or 3 in a polyominoid $\mathbf{P}$. Then v is peripheral.*

Proof. Let $\mathbf{P}= (V, E)$ be a polyominoid and $(x, y) \in V$ an active vertex of degree 2 or 3. By definition, v is situated in a cell, i.e. inside a cycle. Let γ be a maximal cycle having v inside it. If v is on γ, then the proof is complete. Otherwise, there will be a nearest vertex $u = (x', y)$ on γ such that $x' < x$. But $\mathbf{P}$ is a polyominoid and any cycle γ does not contain a hole, i.e, the edges of the segment $[(x', y), (x, y)]$ are in E. This cannot hold, since, v does not admit any edge on its left side.

□

The election algorithm proposed here removes an active vertex once its lifetime has expired. To continue the process, we have to show that the residual graph is still a polyominoid.

Proposition 2. *Let* $\mathbf{P} = (V, E)$ *be a polyominoid of size* ≥ 2 *and let* v *be an active vertex in* $\mathbf{P}$*. The graph* $\mathbf{P}' = (V \setminus \{v\}, E \setminus \{\{v, u\}, u \in V\})$ *is a polyominoid.*

Proof. Let $\mathbf{P}$, v and $\mathbf{P}'$ be as above. To show the proposition, we have to prove that $\mathbf{P}'$ is a connected graph without holes.

- If $deg(v) = 1$, then the suppression of v and its incident edge in $\mathbf{P}$ introduces neither a disconnection nor a hole.
- If $deg(v) = 2$ then let v, v_1, v_2, v_3 be four rectangular vertices of a polyominoid $\mathbf{P}$ such that v is the removable vertex. Consider a vertex $u \in V \setminus \{v, v_1, v_2, v_3\}$. Then, if the vertex u is accessible to a vertex $v_i, 1 \leq i \leq 3$. through a path which passes by v, then when v is removed, u remains accessible to v_i by another path which borrows the vertices $v_{j \neq i}$, $j = 1, 2, 3$. Therefore, by Lemma 1, v is a peripheral vertex and on the other hand its removal generates no hole.
- If $deg(v) = 3$ then the proof is similar to the previous case.

□

3.2. **Standard Spanning Tree.** Let $\mathbf{P}$=(V, E) be a polyominoid. The graph $T = (V, F)$ which traces the weight transmissions in the algorithm is described as follows:

- if $e = \{(x, y), (x + 1, y)\}$ is an edge in E then e belongs to F, i.e. each horizontal edge in E belongs to F:
$$e = \{(x, y), (x + 1, y)\} \in E \Longrightarrow e \in F$$
- if $e = \{(x, y), (x, y + 1)\}$ belongs to E and e is not a left side edge of a cell in $\mathbf{P}$ then e is belong to F, i.e. each vertical edge, who is not a left edge of a cell, belongs to F:
$e = \{(x, y), (x, y + 1)\} \in E \Longrightarrow e \in F$ iff $\{(x, y), (x + 1, y), (x + 1, y + 1), (x, y + 1)\}$ is not a cell in $\mathbf{P}$.

Proposition 3. *The graph* $T = (V, F)$ *described above is a spanning tree of the polyominoid* $\mathbf{P}$*.*

Proof. We can prove this proposition by the inductive construction on $\mathbf{P}$, see [13] for more details. □

Remark. The spanning tree resulting from these rules is unique.

Definition. The spanning tree $T = (V, F)$ is called the *standard spanning tree* of the polyominoid $\mathbf{P}$.

Example 1. Figure 2 gives the standard spanning tree of the polyominoid given in Fig.1.

Proposition 4. *Let* $\mathbf{P} = (V, E)$ *be a polyominoid and* $T = (V, F)$ *be its standard spanning tree. Then, the vertex* $v \in V$ *is active in* $\mathbf{P}$ *iff it is a leaf in* T*.*

Proposition 5. *Let* $\mathbf{P}$ *be a polyominoid of size* ≥ 2*, suppose that* v *is an active vertex in* $\mathbf{P}$ *and let* T *be the standard spanning tree of* $\mathbf{P}$*. Then, let* $\mathbf{P}'$ *denote the residual polyominoid once* v *and its incident edges have been removed and* T' *be the standard spanning tree of* $\mathbf{P}'$*. Then,* T' *can be obtained from* T *by the elimination of the leaf* v *and its incident edge.*

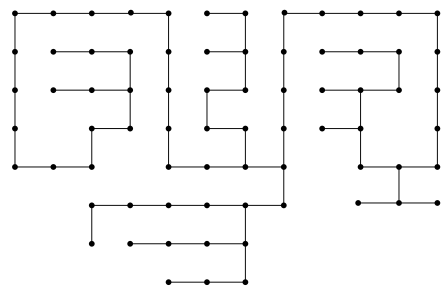

FIGURE 2. Standard spanning tree of the polyominoid given in Fig. 1.

Proof. Let $\mathbf{P}$ be a polyominoid of size ≥ 2, and T its standard spanning tree. Clearly, the residual tree T', after the suppression of a leaf v and its incident edge in T, is a spanning tree of the polyominoid $\mathbf{P}'$ resulting from $\mathbf{P}$ once v and its incident edges are removed. Now, it remains to prove that T' is the standard spanning tree of $\mathbf{P}'$:

- Obviously, the horizontal edges of $\mathbf{P}'$ are in T'.
- The residual vertical edges of $\mathbf{P}'$, which are not situated on the left-hand side of a cell in $\mathbf{P}'$, satisfy the same condition in $\mathbf{P}$ and hence, are in T. Therefore, they are in the standard spanning tree of $\mathbf{P}'$.

□

Putting together the results of this section, we conclude with the following scheme of distributed probabilistic algorithm.

while P is not reduced to a unique vertex
do
- any vertex which active or becomes active (rules $R_0 - R_3$) generates its lifetime according to its weight,
- once the lifetime of an active vertex has expired, it is removed with incident edges and its neighbor in the standard spanning tree collects its weight.

od

4. Analysis of the Algorithm

The election algorithm in a polyominoid is viewed as an election algorithm in its standard spanning tree: as seen in Proposition 4, each active vertex in a polyominoid is a leaf in its standard spanning tree and the weights of the two vertices are equal.

Given $\mathbf{P}=(V,E)$ a polyominoid. Initially, all vertices have the same *weight* 1 : $w(v) = 1, \forall v \in V$. According to the rules seen in section 4, when an active vertex vanishes, its successor collects its weight, adding it to its current weight. At the time t when a vertex v becomes active in a residual polyominoid $\mathbf{P}'$, its weight is the number of vanished vertices on its side. The lifetime delay $L(v)$ for v is a r.v. (random variable) having an exponential distribution of parameter $\lambda(v) = w(v)$:

$$Pr\big(L(v) > t\big) = e^{-\lambda(v)t}, \quad \forall t \geq 0.$$

We say that the death of the active vertex v happens according to a Markovian process with the parameter $\lambda(v)$ equal to its weight $w(v)$. This property is equivalent to the one that the death probability of v in the time interval $[t, t+h]$ is $\lambda(v)h+o(h)$, as $h \to 0$ at any time t, and this independent of what is going on elsewhere and of what happened in the past, the assumption which is in agreement with the distributivity of the algorithm. The random process is a variant of pure death processes which are, in turn, special instances of continuous-time-Markov processes (see [7], Chapter XVII).

Theorem 1. *The strategy described below leads to a* **totally** *fair randomized election: in a polyominoid all vertices have the same probability of being elected.*

The proof of this theorem is complicated and is given in the following section, after some preliminary results have been proved.

Uniformity of the Election. The randomized election can mathematically be modeled by a continuous-time Markov process as follows. The initial state of the process is $\mathbf{P}$ (the whole polyominoid). The set of states $\mathcal{E}_\mathbf{P}$ is the set of all subpolyominoids $\mathbf{Q}= (U, F)$ of $\mathbf{P}= (V, E)$ satisfying: whenever two diagonal vertices (i.e. of the form (x, y) and $(x + 1, y - 1)$ or (x, y) and $(x + 1, y + 1)$) are in $\mathbf{Q}$, then the right vertex $(x + 1, y)$, on the cell containing the vertices, is in U, provided that it is in V.

The following proposition shows that $\mathcal{E}_\mathbf{P}$ is the set of all subpolyominoids of $\mathbf{P}$ which can be reached from $\mathbf{P}$ by a sequence of active-removal vertices (recall that when an active vertex is removed all incident edges are removed as well).

Proposition 6. *A subpolyominoid* $\mathbf{Q}$ *of a polyominoid* $\mathbf{P}$ *can be reached with a positive probability iff* $\mathbf{Q}$ *is in* $\mathcal{E}_\mathbf{P}$.

Proof. Let $\mathbf{Q}$ be a subpolyominoid of $\mathbf{P}$ reachable from $\mathbf{P}$ with a positive probability and prove that $\mathbf{Q} \in \mathcal{E}_\mathbf{P}$. According to the process transition definition, $\mathbf{Q}$ must be obtained from $\mathbf{P}$ by $k-$sequence of active-removal vertices ($0 \leq k < n$). For $k = 0$, the proposition obviously holds. Let it be true for k and prove it for $k + 1$. So, let $\mathbf{Q}$ be obtained from some polyominoid $\mathbf{R}$ of $\mathbf{P}$ by removing an active vertex v and its incident edges. $\mathbf{R}$ is in $\mathcal{E}_\mathbf{P}$ by induction and since, a right most vertex of degree ≥ 2 cannot be active, the resulting subpolyominoid Q will satisfy the condition of being in $\mathcal{E}_\mathbf{P}$.

Let now $\mathbf{Q}= (U, F)$ be in $\mathcal{E}_\mathbf{P}$. we prove by a decreasing induction over $m = |U|$ that $\mathbf{P}$ can be reached by $n - m$ transitions with a positive probability. For $m = n$, $\mathbf{Q}$ is equal to $\mathbf{P}$ and therefore $\mathbf{Q} \in \mathcal{E}_\mathbf{P}$. Suppose that $m < n$, we have to show that there is a vertex $v \in V \setminus U$, such that its addition to U and the addition of all edges with one endpoint v and the other vertex in U, yields a new polyominoid $\mathbf{R}$ belonging to $\mathcal{E}_\mathbf{P}$. Since $m < n$ there is a vertex $u \in V \setminus U$. Consider a path from u to a vertex $s \in U$, let v to be the last vertex of the path which does not belong to U. Then, v has a neighbor vertices in U.

- If v has no other neighbor vertex in U, then clearly, v is an active vertex in $\mathbf{R}$ (its degree is 1 in $\mathbf{R}$). Moreover, $\mathbf{R}$ is in $\mathcal{E}_\mathbf{P}$.
- Otherwise, v has two or three neighbor vertices in U. In this case, let (x, y) be the coordinates of the vertex v. According to this assumption, $\mathbf{Q}$ is in $\mathcal{E}_\mathbf{P}$, there are no neighbor vertices in U, $(x, y+1)$ and $(x-1, y)$ or $(x-1, y)$ and $(x, y - 1)$ such that v is a vertex on the right side of a cell in $\mathbf{R}$ containing the vertices. Consequently, v is a vertex on the left side of one or two cells in $\mathbf{R}$. However, v is an active vertex in $\mathbf{R}$. Moreover, $\mathbf{R} \in \mathcal{E}_\mathbf{P}$.

□

Let $\mathbf{Q}$ be the state of the system at instant t. According to the distributive random structure of the algorithm, any active vertex v of $\mathbf{Q}$ has a lifetime exponentially distributed with a parameter equal to its weight. This is equivalent to the fact that in the time interval $[t, t + \Delta t]$, v may disappear with all incident edges with probability $w(v)\Delta t + o(h)$, as $h \to 0$, and this independent of what is going on elsewhere and what happened in the past.

One can easily show that the probability of passing from $\mathbf{Q}$ to $\mathbf{R}$ is obtained by the removal of active vertex v and its incident edges is given by:

$$P_{(\mathbf{Q},\mathbf{R})} = \frac{w(v)}{\sum_{u \text{ active in } \mathbf{Q}} w(u)} \tag{1}$$

provided that $\mathbf{Q}$ is not reduced to a vertex. The absorbing states (see [7]) are polyominoids reduced to a vertex (the elected vertex).

A mathematical description of probability of being in state $\mathbf{Q}$ at time t can be given as the solution of a system of differential equations. The following proposition can be proved without any difficulty by a straightforward adaptation of the proof given in [7], Chapter XVII, Section 5.

Proposition 7. *Let $\mathbf{Q}$ be in $\mathcal{E}_{\mathbf{P}}$ and let $P_{\mathbf{Q}}(t)$ denote the probability that the state of the election at time t is $\mathbf{Q}$. We have:*

(i) $\dfrac{dP_{\mathbf{P}}(t)}{dt} = -w(\mathbf{P})P_{\mathbf{P}}(t)$,

(ii) *for all subpolyominoid $\mathbf{Q} \neq \mathbf{P}$ of size at least 2 and in $\mathcal{E}_{\mathbf{P}}$,*

$$\frac{dP_{\mathbf{Q}}(t)}{dt} = -w(\mathbf{Q})(t)P_{\mathbf{Q}}(t) + \sum_v w(v)P_{\mathbf{R}}(t),$$

with $\mathbf{R} = \mathbf{Q} \cup (\{v\}, \{\{v,u\}, u \text{ adjacent to } v \text{ in } T\})$, (recall that T is standard spanning tree of $\mathbf{P}$)

where the summation is extended to all vertices v adjacent to $\mathbf{Q}$ in T which do not belong to $\mathbf{Q}$, and

(iii) $\dfrac{dP_{(\{v\},\emptyset)}(t)}{dt} = \displaystyle\sum_{u \text{ adjacent to } v \text{ in } \mathbf{P}} w(u)P_{(\{v,u\},\{\{v,u\}\})}(t)$,

with the initial condition $P_{\mathbf{P}}(0) = 1$.

This proposition characterizes *in principle* the distribution probability of states at a given time t. In particular, it should enable us to compute the absorption probabilities [7]. However, no explicit solution is known to the authors.

Propositions 3-5 allow to confirm that any sequence of transitions over $\mathcal{E}_{\mathbf{P}}$ can be simulated, with the same probability, by a sequence of transitions over the set of factor trees in the standard spanning tree of $\mathbf{P}$ (recall that a factor of a tree is a tree obtained by a sequence of leaf removals). Thus, the study of the process is translated into that of the election over a tree, proposed and analyzed in [19]. In this model, initially all vertices have the same weight 1. Each leaf has a lifetime which is an exponentially distributed random variable with a parameter equal to the weight of the leaf. Once the lifetime of the leaf has expired, it is removed with the incident edge and its weight is recuperated by its father. The process continuous on until the tree is reduced to one vertex, which is considered as the elected vertex. Therefore, the probability of being elected for a vertex v in a polyominoid $\mathbf{P}$ is the same as in the standard spanning tree T and this has been shown to be $\frac{1}{n}$, where n is the size of $\mathbf{P}$.

We enumerate here intermediate results and give the outline of the proof. In the sequel, we suppose that T is the spanning tree of polyominoid $\mathbf{P}$ of size n. Leaves of $\mathbf{P}$ are removed following the random process described above until T is reduced to a unique vertex. We have to prove the uniformity of the chance for all vertices of T.

We first introduce a slight modification of the leaf-removal model over T. We translate the model into a variant on directed trees. For a given vertex v, the unique rooted tree at v can be defined. These rooted trees can be used in a natural way to compute the absorption probabilities.

We consider forests of rooted trees. Let F be a forest of rooted trees, we introduce a death process on F as follows. Each leaf v has an exponentially distributed lifetime with a parameter equal to its weight; initially, all vertices of F are of weight 1. At any time interval $[t, t+\Delta t]$, if the lifetime of a leaf has expired, the leaf is removed with its unique incident edge. If the vanishing leaf has a father, then its father picks up its weight, adding it to its weight. The leaf-removal process goes on the reduced forest until the forest totally disappears.

For a given forest F, let $L(F)$ be the vanishing time; it is a positive-real-valued r.v..

The following proposition is surprising. It asserts that $L(F)$ depends only on the size of the forest and not on its structure.

Proposition 8. *Let F be a forest of size n then the distribution function $G_F(t)$ of the r.v. $L(F)$ is given by:*

$$G_F(t) = Pr(L(F) \leq t) = (1 - e^{-t})^n, \quad \forall t \geq 0.$$

Proof. By induction on n. If F is reduced to a vertex, then the proposition holds (the lifetime for a single vertex is an exponentially distributed r.v. with parameter 1). Suppose that the proposition holds for forests of size less than n and let us prove it for a forest F of size n ($n \geq 2$).

(i) Suppose that F consists of forests $F_1, F_2, \cdots, F_k$ with $k \geq 2$. Let $n = n_1 + n_2 + \cdots + n_k$, where n_i is the size of F_i, $1 \leq i \leq k$. In this case, $L(F_i)$, $1 \leq i \leq k$, are mutually independent r.v. and hence by the induction hypothesis:

$$\begin{aligned} Pr(L(F) \leq t) &= \textstyle\prod_{i=1}^{k} Pr(L(F_i) \leq t) \\ &= \textstyle\prod_{i=1}^{k} (1 - e^{-t})^{n_i} \\ &= (1 - e^{-t})^n. \end{aligned}$$

(ii) Otherwise, suppose that F has size n and consists of a unique root r and rooted trees $A_1, A_2, \cdots, A_k$. Now, let F' be the forest consisting of A_1, $A_2, \cdots, A_k$ (alternatively, let $F' = A_1 \cup \cdots \cup A_k$). F' has size $n-1$ and, by the induction hypothesis, $(1-e^{-t})^{n-1}$ is the distribution function of $L(F')$. But, $L(F)$ is the sum of two independent r.v. $L(F')$ and the lifetime of r. The last one is an exponential r.v. of parameter n (weight of r). Thus, $L(F)$ has the distribution function (see [8], p. 142. Theorem 2) given by:

$$\begin{aligned} G_F(t) &= \textstyle\int_0^t G_{F'}(t-x) d(1 - e^{-nx}) \\ &= \textstyle\int_0^t G_{F'}(t-x) n e^{-nx} dx, \end{aligned}$$

where $G_{F'}(t-x) = (1 - e^{t-x})^{n-1}$.

Hence:

$$\begin{aligned} G_F(t) &= \int_0^t n[1 - e^{-(t-x)}]^{n-1} e^{-nx} dx \\ &= \int_0^t n[e^{-x} - e^{-t}]^{n-1} e^{-x} dx \\ &= [-(e^{-x} - e^{-t})^n]_{x=0}^{x=t} \\ &= (1 - e^{-t})^n. \end{aligned}$$

The proposition follows. □

Given two forests F_1 and F_2, we say F_1 *beats* F_2, if $L(F_1) \geq L(F_2)$. The next result easily follows from the above lemma.

Corollary 1. *Let F_1 and F_2 be two forests of sizes n_1 and n_2 respectively. The probability that F_1 beats F_2 is $\frac{n_1}{n_1+n_2}$.*

Lemma 2. *Consider a vertex v in T with the adjacent vertices $v_1, \ldots, v_k$. Let F consist of two trees A and B obtained by the suppression of edge $\{v, v_1\}$ rooted at v and v_1 respectively. Then, the probability that v is removed before the whole tree factor on the side of v_1 (i.e. undirected B) in the election process over T is the same as the probability of v_1 beating v in F.*

Proof. The events whose probabilities are to be calculated can be represented as sequences of leaves being removed:

- $\sigma = \langle l_1, \ldots, l_k \rangle$, where $l_i, 1 \leq i \leq k$ are leaves or vertices which become leaves in T (or in F respectively) after the removal of some previous vertices in the sequence,
- $l_k = v$ and
- v_1 does not figure in σ.

On the one hand, it is easy to see that any sequence satisfying the above conditions in T does it in F and vice versa. On the other hand, the probability of such σ according to (1) is:

$$P(\sigma) = \prod_{1 \leq i \leq k} q_i,$$

with

$$q_i = \frac{\lambda(l_i)}{\lambda(T_i)},$$

where T_i is the residual tree (arborescence respectively) just before the l_i removal. In each step of the leaf removal along σ, T and F have the same set of leaves and, hence, the involved quantities on T are the same as the corresponding ones on F. The lemma follows. □

Proposition 9. *Let $q(v)$ denote the probability of being elected in T for a vertex v. We have $q(v) = \frac{1}{n}$.*

Proof. For $n = 1$ or $n = 2$ the proposition is obvious. Otherwise, let $v_1, \ldots, v_k$ be the adjacent vertices to v. Let, on the other hand, $A_1, \ldots, A_k$ be disjoint tree rooted at $v_1, \ldots, v_k$ of sizes $n_1, \ldots, n_k$ respectively. Clearly, v fails iff it vanishes before one of the factors situated on the v_i side for $1 \leq i \leq k$. These last events are pairwise disjoint and therefore, according to the previous lemma, the failure probability of v is the sum of the probabilities of v being beaten by one of its neighbors v_i in the forest consisting of the tree rooted at v and v_i respectively. Hence, according to Corollary 1, we have:

$$1 - q(v) = \sum_{i=1}^{k} \frac{n_i}{n}.$$

Since, $\sum_{i=1}^{k} n_i = n - 1$, the proposition follows. □

Proof of Theorem 1. Straightforward by the similarity of the election process over **P** and over its standard spanning tree T and the previous proposition. □

References

[1] D. Angluin. Local and global properties in networks of processors. In *Proceedings of the 12th Symposium on theory of computing*, pages 82–93, 1980.

[2] M. Bousquet-Mélou. Convex polyominoes and heaps of segments. *J. Phys. A: Math. Gen.*, 25:1925–1934, 1992a.

[3] M. Bousquet-Mélou. A method for the enumeration of various classes of column-convex polygons. *Discrete Mathematics*, 154:1–25, 1996.

[4] A. Bretto. *Introduction to Hypergraph Theory and its Applications to Image Processing. Mongraphy in: Advances in Imaging and Electron Physics*. Academic Press, March 2004.

[5] M. Delest, J.-P. Dubernard, and I. Dutour. Parallelogram polyominoes and corners. *J. Symbolic Comput.*, 20:503 –515, 1995.

[6] M.-P. Delest and G. Viennot. Algebraic languages and polyominoes [sic] enumeration. 34:169–206, 1984.

[7] W. Feller. *An Introduction to Probability Theory and Its Application, Volume I*. John Wiley and Sons, 1950.

[8] W. Feller. *An Introduction to Probability Theory and Its Application, Volume II*. John Wiley and Sons, 1966.

[9] M. Gardner. *Mathematics, magic and mystery*. New York. Dover Publications, 1956.

[10] M. Gardner. *New mathematical diversions*. Washington. Math. Assoc. Amer., 1995.

[11] S.W. Golomb. Checker boards and polyominoes. *American Mathematical Monthly*, 1954.

[12] S.W. Golomb. *Polyominoes: puzzles, patterns, problems, and packings*. Princeton University Press, 1996.

[13] A. El Hibaoui, N. Saheb-Djahromi, and A. Zemmari. Polyominoids and uniform election. Technical Report 1337-04, LaBRI, Université Bordeaux 1, November 2004.

[14] G. Le Lann. Distributed systems – toward a formal approach. In *Proceedings of the IFIP Congress 77*, pages 155–160, 1977.

[15] I. Litovsky, Y. Métivier, and W. Zielonka. On the recognition of families of graphs with local computations. *Inform. and Comput.*, 118:110–119, 1995.

[16] A. Del Lungo, A. Frosini, M. Nivat, and L. Vuillon. Discrete tomography: reconstruction under periodicity constraints. In P. Widmayer, F. Triguero, R. Morales, M. Hennessy, S. Eidenbenz, and R. Conejo, editors, *ICALP*, volume 2380 of *Lecture Notes in Computer Science*, pages 38–56. Springer, 2002.

[17] A. Mazurkiewicz. Solvability of the asynchronous ranking problem. In *Inf. Processing Letters*, volume 28, pages 221–224, 1988.

[18] Y. Métivier and N. Saheb. Medians and centres of polyominos. *Inform. Proc. Letters*, 57:175–181., 1996.

[19] Y. Métivier, N. Saheb-Djahromi, and A. Zemmari. A uniform randomized election in trees (extended abstract). In *Proceedings of The 10th International Colloquium on Structural Information and Communication Complexity (SIROCCO 10)*, pages 259–274. Carleton university press, 2003.

[20] M. Raynal. *Networks and distributed computation : concepts, tools, and algorithms*. MIT Press, Cambridge, 1988.

[21] K. H. Rosen. *Handbook of discrete and combinatorial mathematics*. CRC Press, 2000.

[22] Robert Sedgewick. *Algorithms in C++*. Addison-Wesley Co., 1st edition, 1992.

[23] G. Tel. *Introduction to distributed algorithms*. Cambridge University Press, 2000.

[24] H.N.V. Temperley. Combinatorial problems suggested by the statistical mechanics of domains and of rubber-like molecules. *Phy.*, 103:1–16, 1956.