Chip-firing games on directed graphs

Anders Björner
Department of Mathematics,
Royal Institute of Technology
S-100 44 Stockholm, Sweden

László Lovász
Department of Computer Science,
Eötvös Loránd University,
Budapest, Hungary H-1088
and
Princeton University, Princeton, NJ 08544

(Revised, July 1992)

Abstract. We consider the following (solitary) game: each node of a directed graph contains a pile of chips. A move consists of selecting a node with at least as many chips as its outdegree, and sending one chip along each outgoing edge to its neighbors. We extend to directed graphs several results on the undirected version obtained earlier by the authors, P. Shor, and G. Tardos, and we discuss some new topics such as periodicity, reachability and probabilistic aspects.

Among the new results specifically concerning digraphs, we relate the length of the shortest period of an infinite game to the length of the longest terminating game, and also to the access time of random walks on the same graph. These questions involve a study of the Laplace operator for directed graphs. We show that for many graphs, in particular for undirected graphs, the problem whether a given position of the chips can be reached from the initial position is polynomial time solvable.

Finally, we show how the basic properties of the "probabilistic abacus" can be derived from our results.

0. Introduction

Let G be a directed graph and let us place a pile of chips on each node of G. We are allowed to change this arrangement of chips as follows: we select a node which has at least as many chips as its outdegree, and move one chip along each outgoing edge to the neighbor at the other end. We call this step *firing* this node. We can repeat this as long as we find some node that can be fired. A (finite or infinite) sequence of firings is called a *chip-firing game.* The sequence of points fired is called the *record* of the game.

There are a number of natural questions to ask: will this procedure be finite or infinite? If finite, how long can it last? If infinite, how soon can it cycle? What role is played by the choices that are made? How can one determine if a given distribution of chips can be transformed into another given one by firings?

Chip-firing games were introduced, independently, at least twice. This is not counting the relation to general Petri nets, of which they are (untypical) special cases, or the obvious similarity with neural nets, which remains unexplored.

Engel [7,8] considered a chip-firing procedure he called the "probabilistic abacus", as a method of determining the absorbtion probabilities and access times of certain Markov chains by combinatorial means. As we shall show in Section 6, the probabilistic abacus may be viewed as a special case of chip firing in our sense, and its basic properties are easy to derive in this setting.

Spencer [20] introduced a diffusion-like process on the line as a tool in analyzing a certain "balancing" game. His process may be viewed as chip firing on a very long undirected path. Anderson, Lovász, Shor, Spencer, É. Tardos and Winograd [3] studied the process in greater detail, and observed, among other things, the key property of independence of the length of the game, and of the final position, from the special choices made during the process. The procedure was extended from paths to general graphs by Björner, Lovász and Shor [4], who especially studied the undirected case. G. Tardos [21] proved that if an undirected game terminates at all, then it terminates in a polynomial number of steps, while Eriksson [10] showed that a terminating directed game can be exponentially long.

The analysis of chip-firing games in this paper relies on two components. One is the formal language approach developed in [4]: The records of all possible chip-firing games from a given initial position give a language with rather strong exchange properties. The other is the analysis of the Laplace operator of the directed graph, and its close connections with chip firings and random walks on the graph. The connection with the Laplace operator in the undirected case was already observed in straightforward since the Laplace operator is not symmetric any more and we loose its spectral decomposition as a tool. As an application of these methods, we show that a terminating game on a strongly connected graph is not longer than a polynomial in its size times the length of the period of a periodic game. This in a sense extends Tardos's theorem to the directed case.

We describe an algorithm to decide if one position of chips can be transformed into another given position by a sequence of firings. The algorithm is polynomial in the case of undirected graphs without multiple edges, but exponential in general. The analysis also implies that if a position on an undirected graph without multiple edges can be transformed into another one, then it can be so transformed by a polynomial number of moves. This does not remain true for directed graphs.

Chip-firing games are special cases of so called "vector addition systems" (defined in Section 1), and some of our results can be generalized to this setting. In particular, the reachability algorithm for chip positions (Section 5) improves some instances of the decidability result of Kosaraju [14] and Mayr [15] for vector addition systems.

A different kind of chip-firing game, motivated by probabilistic and algebraic considerations, has recently been introduced by Diaconis and Fulton [6]. A common generalization of the game described here and that of Diaconis and Fulton appears in Eriksson [9].

We are grateful to N. Alon and P. Diaconis for helpful comments on an early draft of this paper, and to a referee for suggesting improvements.

Preliminaries. All graphs we consider are finite and directed. Loops and multiple edges are allowed. An undirected graph is identified with the directed graph obtained by replacing each edge between nodes i and j by a pair of oppositely directed edges between i and j.

For a digraph (directed graph) $G = (V, E)$ we let $n = |V|$ and $m = |E|$. We denote by $d^+(k)$ and $d^-(k)$ the outdegree and indegree of the node k, respectively (where the digraph $G = (V, E)$ is fixed), and by $d_{i,j}$ the number of edges from i to j. Thus, $d^+(k) = \sum_j d_{k,j}$, $d^-(k) = \sum_i d_{i,k}$, and $d_{i,i}$ is the number of loops at node i. Finally, we let $D = \max_{k\in V} d^+(k)$. Clearly, $D \le m$; and if G lacks multiple edges then $D \le n$.

For every vector $v \in \mathbb{R}^V$, we denote

$$|v| = \sum_{k\in V} |v_k| \qquad \text{and} \qquad \|v\| = \sum_{k\in V} d^+(k)|v_k|.$$

A digraph is *eulerian* if $d^+(k) = d^-(k)$ for every node k. Every undirected graph is eulerian, and we shall see that many of the results (but not all of the methods) extend from the undirected to the eulerian case without any change.

A directed graph is *strongly connected* if there is a directed path from i to j, for every ordered pair of nodes i and j. A connected eulerian graph is strongly connected. A general digraph has a uniquely induced partition into strongly connected components. A strongly connected component which is not connected to the outside by any edge leaving the component will be called a *sink component.*

1. Chip firing, vector addition systems, and exchange languages

The following basic fact about chip-firing games was established in [4, Theorem 2.1 and Remark 3.4].

1.1 Theorem *Given a directed graph G, and an initial distribution of chips, either every legal game can be continued indefinitely, or every legal game terminates after the same number of moves with the same final position. The number of times a given node is fired is the same in every legal game.*

The original proof was based on a formal language interpretation of chip-firing games. This approach will be reviewed in this section, since we will make substantial use of it again in this paper.

A simpler and more direct proof was found by Eriksson [9]. Eriksson actually proves a more general version of Theorem 1.1, where only the outdegree and the number of loops at each node are fixed in advance (so that one knows when a firing is legal and how many chips should then be moved), and where each time a node has been fired the graph may be freely redrawn within the given outdegree constraints. We will not use this added degree of generality in this paper, but remark that it might be of interest for the study of Markov chains with non-stationary transition probabilities.

Let V be a finite set. A *language* over V is any set of finite strings (or words) formed by elements of V. A *subword* of a word α is any string obtained by deleting letters from α arbitrarily. We denote by $|\alpha|$ the length of the word α and by $[\alpha]$ the *score* of α, i.e., the vector $a \in \mathbb{Z}_+^V$ whose i-th entry is the number of times letter i occurs in α. We denote by $|a|$ the ℓ_1-norm (sum of absolute values of entries) of the vector a. Thus $|[\alpha]| = |\alpha|$.

A word in a language is called *basic* if it is not the beginning section of any other word.

A language $\mathcal{L}$ is called *left-hereditary* if whenever it contains a string, it contains all beginning sections of the string. The language is called *permutable* if whenever $\alpha, \beta \in \mathcal{L}$, $[\alpha] = [\beta]$, and $\alpha x \in \mathcal{L}$ for some $x \in V$, we also have $\beta x \in \mathcal{L}$. Finally, we say that $\mathcal{L}$ is *locally free* if whenever $\alpha x \in \mathcal{L}$ and $\alpha y \in \mathcal{L}$ for two distinct elements $x, y \in V$, we also have $\alpha x y \in \mathcal{L}$.

Locally free permutable left-hereditary languages have many nice properties. The following proposition summarizes some of the results from Section 2 of [4].

1.2 Proposition *Let $\mathcal{L}$ be a locally free permutable left-hereditary language. Then:*

(i) *If $\alpha, \beta \in \mathcal{L}$ then there exists a subword α' of α such that $\beta\alpha' \in \mathcal{L}$ and $[\beta\alpha']$ is the entry-wise maximum of $[\alpha]$ and $[\beta]$.*

(ii) *If there is a basic word then the language is finite.*

(iii) *Every basic word has the same score (in particular, the same length).*

(iv) *If $\mathcal{L}$ is finite then two words $\alpha, \beta \in \mathcal{L}$ have $[\alpha] = [\beta]$ if and only if*

$$\{\gamma : \ \alpha\gamma \in \mathcal{L}\} = \{\gamma : \ \beta\gamma \in \mathcal{L}\}.$$

∎

Property (i), called the *strong exchange property,* expresses that the game is an *antimatroid with repetition*; for a discussion of this connection, see [4] or [5].

The following lemma, whose proof is trivial and omitted (see [4, Lemma 2.4]), shows that these general results can be applied to chip-firing games.

1.3 Lemma. *The records of legal games on a digraph starting from a fixed initial position form a locally free permutable left-hereditary language.* ∎

It is clear that the score of a word in this language determines the position reached by the corresponding game, and that such a word is basic if and only if that position is terminal. Therefore Theorem 1.1 is a direct consequence of Proposition 1.2.

We will later need to consider some related languages associated with digraphs. For a language $\mathcal{L}$ over alphabet V, and for a vector $a \in \mathbb{Z}_+^V$, let $\mathcal{L}[a]$ denote the set of those words in $\mathcal{L}$ that contain letter i at most a_i times. It is straightforward to verify the following.

1.4 Lemma. *If $\mathcal{L}$ is a locally free permutable left-hereditary language then so is $\mathcal{L}[a]$.*

In connection with chip-firing games this means that the convergence property expressed by Theorem 1.1 remains valid if there is a capacity $a_i \in \mathbb{Z}_+ \cup \{\infty\}$ associated with each node i, and node i is allowed to be fired at most a_i times in any legal game.

We conclude this section with describing a more general construction leading to locally free permutable left-hereditary languages. This leads to discussing a real-valued version of the chip-firing game.

Let C be a convex cone in $\mathbb{R}^n$ (with apex in 0), and let V be a finite set of vectors in $\mathbb{R}^n$. Let b be any fixed vector in C. Let $\mathcal{L}$ be the set of all sequences $v_1 \dots v_k$ of vectors in V for which $b + v_1 + \ldots + v_i \in C$ for every $1 \le i \le k$. We call (C, V, b) a *vector addition system* and $\mathcal{L}$ a *vector addition language.* Vector addition systems were introduced by Karp and Miller [11] with the further requirement that b and the vectors in V have integer components and that C is the first orthant of $\mathbb{R}^n$. We will call such vector addition systems *integral.* They are also known as "general Petri nets", see Reisig [17] or Reutenauer [18].

It is obvious that every vector addition language is left-hereditary and permutable, but in general not locally free. However, let us assume that the following holds:

$(*)$ *for every facet $a^T x \ge 0$ of C, there exists at most one vector $v \in V$ with $a^T v < 0$.*

1.5 Proposition *If V and C have property $(*)$, then the vector addition language defined by them starting at any $b \in C$ is locally free.*

Proof. Let $v_1 \dots v_k u \in \mathcal{L}$ and $v_1 \dots v_k w \in \mathcal{L}$. Assume that $v_1 \dots v_k u w \notin \mathcal{L}$, then we must have that $b + v_1 + \ldots + v_k + u + w \notin C$. So there exists a facet $a^T x \ge 0$ of C which is violated by $b + v_1 + \ldots + v_k + u + w$. But since this facet is satisfied by both $b + v_1 + \ldots + v_k + u$ and $b + v_1 + \ldots + v_k + w$, we must have $a^T u < 0$ and $a^T w < 0$, which contradicts assumption $(*)$. ∎

Any chip-firing language on a loop-free digraph is a special case of an integral vector addition language, where C is the non-negative orthant in $\mathbb{R}^V$, and for every node i we take the vector v_i defined by

$$(v_i)_j = \begin{cases} d_{i,j}, & \text{if} \quad j \ne i, \\ -d^+(i), & \text{if} \quad j = i. \end{cases}$$

Clearly this vector system satisfies $(*)$.

The definition of a vector addition language and of property $(*)$ can easily be generalized so that chip-firing languages of digraphs *with loops* are special cases and Proposition 1.5 remains valid. We leave the details to the reader.

We now describe the real-valued analogue of the chip-firing game, which can be called the *mass-firing game.* Here we have a fixed underlying digraph G with at most one edge from i to j, for every pair of nodes i and j, and a fixed weight function $(i,j) \in E \mapsto d_{i,j} \in \mathbb{R}_+$. As initial position we have a real mass distribution on the nodes $p : V \to \mathbb{R}_+$. If $p_i \geq d^+(i) = \sum d_{i,j}$ then it is legal to fire node i, meaning that a new position is created by removing $d^+(i)$ from the mass at i and adding $d_{i,j}$ to the mass of each neighbor j. A game is played by successive choices of legal moves. Propositions 1.2 and 1.5 show that the $\mathbb{R}$-valued mass-firing game satisfies the basic convergence property expressed by Theorem 1.1.

A $\mathbb{Z}$-valued mass-firing game is clearly equivalent to our chip-firing game: just replace each weighted edge $d_{i,j} \in \mathbb{Z}_+$ by that many parallel edges from i to j, and each mass $p_i \in \mathbb{Z}_+$ by that many chips. A $\mathbb{Q}$-valued mass-firing game is equivalent to a $\mathbb{Z}$-valued one by scaling. However, $\mathbb{R}$-valued mass-firing games are truly more general than chip-firing games. The reason that we have chosen to remain in the $\mathbb{Z}$-valued setting in this paper is (in addition to the intuitive appeal of the chips model) that the analysis of periodicity, which plays an important role here, would otherwise be more complicated. The $\mathbb{R}$-valued mass-firing games might be of interest for the study of Markov chains with irrational transition probabilities.

2. Finite termination

Let us recall from [4] the following facts about chip-firing games on undirected graphs, which are very helpful in distinguishing finite and infinite games in the undirected case: (i) [4] *if a game is infinite, then every node gets fired infinitely often;* (ii) [20] *if a game is finite, then there is a node that is never fired.* Now, (i) extends to digraphs in the following form without any difficulty. See Proposition 4.4 and Corollary 4.5 for an extension of (ii).

2.1 Lemma. *In every infinite game on a digraph, every node in some sink component is fired infinitely often.*

Proof. If there is an edge from k to j, and node k is fired infinitely many times then so is j (otherwise an infinite number of chips would build up on j after it has stopped firing). Hence, if k is fired infinitely often then so are all nodes reachable from k. Since some node k must be fired infinitely often and some sink component is reachable from k, this completes the proof. ∎

It should be observed that a *sink* (a node k with $d^+(k) = 0$) according to the rules can be fired at any time. This gives a degenerate firing which does not affect the current position. It is often natural to view sinks as absorbing nodes that can never be fired, and this can easily be achieved in our model by attaching sufficiently many loops to them.

Given a graph, we may ask: what is the minimum number of chips that allows an infinite game? What is the maximum number of chips that allows a finite game? In [4] it was shown that for an undirected graph with n nodes and m (undirected) edges, more than $2m - n$ chips guarantee that the game is infinite; fewer than m chips guarantee that the game is finite; and for every number N of chips with $m \leq N \leq 2m - n$, there are initial positions that lead to an infinite game and initial positions that lead to a finite game.

For directed graphs, one of the above questions can still be answered trivially: if G is a directed graph with n nodes and m edges, and we have $N > m - n$ chips, then the game is infinite (there is always a node that can be fired, by the pidgeon hole principle), and $N \le m - n$ chips can be placed so that the game terminates in 0 steps.

We don't know how to determine the minimum number of chips allowing an infinite game on a general digraph. This is not a function just of the number of nodes and edges, even if the graph is eulerian. Consider a circuit of length n with doubled edges having symmetric orientation. By the "undirected" result, it takes n chips to make an infinite game. But if we reverse the orientation of half of the edges so that we get an oriented circuit with every edge doubled, then 2 chips placed on the same node start an infinite game.

The following bound can, however, be obtained.

2.2 Theorem. *Let G be a strongly connected digraph and let h denote the maximum number of edge-disjoint directed cycles in G. Then any game with fewer than h chips is finite.*

(Note that for the undirected case, this gives the exact result.)

Proof. The proof extends an idea of Tardos [21]. Let $C_1, \dots, C_h$ be a maximum family of edge-disjoint directed cycles. While playing the game, let each cycle "capture" the chip that first uses any of its edges; from then on, this chip has to move along the edges of the cycle. This does not conflict with the game: when we fire a node, we make sure that those chips which are captured should move along the right edges, and this is possible since the cycles are edge-disjoint.

Now, by $N < h$ there will be a cycle that does not capture any chip. This means that the nodes of that cycle are never fired; but by Lemma 2.1, this means that the game is finite. ∎

A version of Theorem 2.2 for general digraphs is obtained by letting h be maximal such that every sink component of G has h edge-disjoint directed cycles.

3. The directed Laplace operator and random walks

Let $G = (V, E)$ be a directed graph and let $L \in \mathbb{R}^{V\times V}$ be the matrix defined by

$$L_{ij} = \begin{cases} d_{j,i}, & \text{if} \quad i \ne j, \\ -d^+(i) + d_{i,i}, & \text{if} \quad i = j. \end{cases}$$

We call L the *Laplacian* of the digraph G. Note that $L^T\mathbf{1} = 0$, so L is singular. If G is eulerian then also $L\mathbf{1} = 0$.

The Laplace operator of an undirected graph has received a considerable amount of attention in connection with the study of expansion properties of graphs and the related mixing properties of random walks on them, see e.g. Alon [2]. The directed case is less studied.

For the analysis of the periodic properties of chip firing games in the next section we will need a description of the null space of L, i.e., the space of all $v \in \mathbb{R}^V$ such that $Lv = 0$. We will show that this space has a basis of non-negative vectors whose supports are the sink components of G. For instance, if G is acyclic then the null space of L is precisely the set of all vectors supported by sink nodes.

3.1 Proposition. *Suppose that $G = (V, E)$ has k sink components $S_1, \dots, S_k$. Then there exist vectors $v_1, \dots, v_k \in \mathbb{Z}^V$ such that*

(i) *$\{v_1, \dots, v_k\}$ is a basis of the null space of the Laplacian of G,*

(ii) *$(v_i)_j > 0$ for $j \in S_i$, and $(v_i)_j = 0$ for $j \notin S_i$,*

(iii) *$(v_i)_j = 1$ for all $j \in S_i$, if S_i is eulerian.*

Proof. First assume that G is strongly connected. If D is the maximum outdegree of G, then $L + DI$ is a non-negative irreducible matrix, which has an all-positive left eigenvector $\mathbf{1}$ with eigenvalue D. Hence, by the Perron-Frobenius Theorem (see e.g. Minc [16]), D is the largest eigenvalue of $L + DI$. Again, by the Perron-Frobenius Theorem, the right eigensubspace of $L + DI$ belonging to D is one-dimensional and spanned by an all-positive eigenvector. But this eigensubspace is just the null space of L. Hence L has rank $n-1$ and its null space is spanned by an all-positive vector, which (after scaling) can be assumed to have integer components. In particular, if G is a connected eulerian digraph then the null space of L consists of the vectors $t\mathbf{1}$, $t \in \mathbb{R}$.

For a general digraph G we claim that no vector v with $Lv = 0$ can have a non-zero entry outside the sink components. Assume that this is not so, and let $T^+ \neq \emptyset$ [resp. T^-] be the set of nodes with positive [resp. negative] entry in $V \backslash (S_1 \cup \ldots \cup S_k)$, for some vector v in the null space. Consider the equation

$$\sum_{j \in V} d_{j,i} v_j = d^+(i) v_i$$

(which is equivalent to $Lv = 0$), and sum it for all $i \in T^+$:

$$\sum_{i \in T^+} d^+(i) v_i = \sum_{i \in T^+} \sum_{j \in V} d_{j,i} v_j = \sum_{j \in V} v_j \sum_{i \in T^+} d_{j,i}. \tag{3.1}$$

On the right hand side it sufficies to sum over $j \in T^+ \cup T^-$, since for $j \notin T^+ \cup T^-$ either $v_j = 0$ or else j is contained in a sink component and there is no edge from j to $i \in T^+$. So we obtain

$$\sum_{j \in V} v_j \sum_{i \in T^+} d_{j,i} = \sum_{j \in T^+} v_j d(j, T^+) + \sum_{j \in T^-} v_j d(j, T^+),$$

where $d(A, B)$ is the number of edges with tail in A and head in B. On the left hand side we obtain

$$\sum_{i \in T^+} d^+(i) v_i = \sum_{i \in T^+} v_i d(i, T^+) + \sum_{i \in T^+} v_i d(i, V \backslash T^+),$$

and hence equation (3.1) implies

$$\sum_{j\in T^-} v_j d(j,T^+) = \sum_{i\in T^+} v_i d(i, V\backslash T^+).$$

Here the left hand side is non-positive and the right hand side is non-negative, so they must both be equal to zero. But this implies that $d(i, V\backslash T^+) = 0$ for every $i \in T^+$, and hence T^+ contains a sink component, which contradicts its definition.

Since no edge connects two distinct sink components, the restriction of any vector in the null space of L to any sink component is in the null space of the Laplacian of that component. This component null space is, as we already showed, one-dimensional and spanned by an all-positive vector. Let $v_1, \ldots, v_k$ be all-positive vectors spanning the null spaces of the Laplacians of the sink components $S_1, \ldots, S_k$. To make them uniquely defined, it will be convenient to scale the v_i so that they have integral coordinates with no common divisor. Abusing notation, we extend the vector v_i outside S_i by adding 0 entries so that it becomes an element of $\mathbb{Z}^V$. The family $\{v_1, \ldots, v_k\}$ now verifies all claims.∎

Remark: For a general non-negative matrix M with largest eigenvalue E, the Perron-Frobenius Theorem [16] gives the existence of *some* non-negative right eigenvector associated with E. The proof method used for Proposition 3.1 can be adapted to show that if **1** is a left E-eigenvector, then there is a *basis* of the right eigensubspace of M belonging to E that consists of non-negative vectors with pairwise disjoint supports.

There is an intrinsic connection between the Laplacian and random walks on digraphs. We discuss this connection here not only because random walks are dynamic processes intimately linked to chip firing, but also because techniques from the theory of random walks will be used in proving the bound in Proposition 3.6 below. For general facts concerning discrete stationary Markov chains, see e.g. Kemeny and Snell [12] or Kemperman [13].

A *random walk* on a digraph G is a Markov chain $y_0, y_1, \ldots,$ assuming values from $V(G)$, so that given the value of y_t, the probability that $y_{t+1} = u$ is proportional to the number of edges connecting y_t to u. (We assume that $d^+(i) > 0$ for every node i.) We shall not go into the theory of random walks, which is particularly well developed in the case of undirected graphs; we shall restrict ourselves to a trivial and a slightly less trivial connection with the Laplacian.

A *stationary distribution* of the random walk on G is a probability distribution on $V(G)$ such that if y_t has this distribution then so does y_{t+1}. The following lemma is a straightforward reformulation of this definition.

3.2 Lemma. *A probability distribution $(q_i : i \in V(G))$ is stationary if and only if the vector x defined by $x_i = q_i/d^+(i)$ satisfies $Lx = 0$.* ∎

In particular, if G is a strongly connected digraph then there is a unique stationary distribution on G, namely $q_i = d^+(i) \cdot x_i$, where x is the unique positive solution of the equations $Lx = 0$, $\|x\| = 1$. Note that $q_i > 0$ for all nodes i. If G is eulerian then substitution shows that this unique vector is defined by $x_i = 1/m$, and so $q_i = d^+(i)/m$, where m is the total number of edges.

The *access time* (or, first passage time) $\mathrm{acc}(i,j)$ of node j from node i is the expected number of steps in a random walk starting at i before it hits j. We denote by $\mathrm{acc}(G)$ the maximum access time between any two nodes. If G is strongly connected then this number is finite. In fact, the following upper bound can be derived by extending an old argument of Aleliunas, Karp, Lipton, Lovász and Rackoff [1]:

3.3 Proposition. *Let G be a strongly connected digraph with stationary distribution q. Let i and j be two nodes connected by a directed path $i = i_0, i_1, \ldots, i_h = j$. Then*

$$\mathrm{acc}(i,j) \le \left(\sum_{r=0}^{h-1} \frac{d^+(i_r)}{q_{i_r}} \right) - h.$$

Proof. Let $e_1 = i_0 i_1, \ldots, e_h = i_{h-1} i_h$ be the successive edges of the given path. Consider a random walk. Assume that we have to wait τ_1 steps to see it pass the edge e_1; after that, we have to wait τ_2 steps to see it pass the edge e_2, etc. So after $\tau_1 + \ldots + \tau_h$ steps, the walk will be at node j, and hence $E(\tau_1 + \ldots + \tau_h)$ is an upper bound on the access time.

By linearity of expectations, $E(\tau_1 + \ldots + \tau_h) = E(\tau_1) + \ldots + E(\tau_h)$. It therefore suffices to show that

$$E(\tau_{r+1}) \le \frac{d^+(i_r)}{q_{i_r}} - 1.$$

We show this for $r = 0$. If $d^+(i) = 1$ then clearly $E(\tau_1) = 1$, and $q_{i_0} \le q_{i_1}$ implies that $q_i = q_{i_0} \le 1/2$. Set $d = d^+(i) > 1$ and $q = q_i$. Starting from i, let $E(k)$ denote the expected number of steps we have to make before seeing i again, if we start through one of the edges from i to k. Then $(1/d)\sum_k d_{i,k} E(k)$ is the expected number of steps starting from i until we return, which is clearly $1/q$ (see [12, Theorem 4.4.5]). Hence the expected number of steps before return, given that we don't start through e_1, is

$$\frac{1}{d-1}\left[\sum_k d_{i,k} E(k) - E(i_1)\right] \le \frac{d}{d-1}\frac{1}{q} - \frac{2}{d-1} = \frac{d-2q}{(d-1)q}.$$

Now the probability that we will pass through e_1 after the t-th visit is $(1/d)\bigl(1-(1/d)\bigr)^t$, and hence the expected time for passing through e_1 is at most

$$\sum_{t=0}^{\infty} \frac{1}{d}\left(1 - \frac{1}{d}\right)^t \left(t\frac{d-2q}{(d-1)q} + 1 \right) = \frac{d}{q} - 1.$$

This proves the assertion. ∎

3.4 Corollary. $\mathrm{acc}(G) \le \|(1/q_k)\|$.

Proof.

$$\mathrm{acc}(i,j) \le \sum_{r=0}^{h-1} \frac{d^+(i_r)}{q_{i_r}} \le \sum_{k\in V} \frac{d^+(k)}{q_k} = \| \left(\frac{1}{q_k}\right) \|.$$

∎

The next lemma expresses another connection between the access time and the Laplacian.

3.5 Lemma. *Let G be a strongly connected digraph and $i,j \in V(G)$. Then there exists a vector $w \in \mathbb{R}^V$ such that*

$$Lw = e_j - e_i, \qquad w \ge 0, \qquad w_j = 0, \tag{3.2}$$

and

$$\mathrm{acc}(i,j) = \|w\|$$

(Note that conditions (3.2), and in fact even the weaker conditions

$$Lw = e_j - e_i, \qquad \min_k w_k = 0,$$

determine w uniquely.)

Proof. Let u_k denote the expected number of times a random walk starting at node i leaves node k before hitting node j. (In particular, we have $u_j = 0$.) Then for every node $r \ne i,j$ we have

$$\sum_k d_{k,r}\, u_k/d^+(k) = u_r,$$

since $u_k/d^+(k)$ is the expected number of times the random walk passes through any one particular edge from k to r. For $r = i$ and $r = j$ we are off by exactly one at the beginning and at the end, respectively, and hence

$$\sum_k d_{k,i}\, u_k/d^+(k) = u_i - 1, \qquad \sum_k d_{k,j}\, u_k/d^+(k) = u_j + 1 = 1.$$

Hence $w_i = u_i/d^+(i)$ satisfies $Lw = e_j - e_i$, and trivially $w \ge 0$ and $w_j = 0$. Moreover, $\|w\| = \sum_k u_k$ is the expected number of steps a random walk starting at i makes before it hits j, i.e., the access time. ∎

Now we formulate the result, needed in the next section, for which this detour through random walks was made.

3.6 Proposition. *Let G be a strongly connected digraph and let $w \in \mathbb{R}^V$ with $\min_k w_k = 0$. Then*

$$\|w\| \le \frac{1}{2}|Lw|\ \mathrm{acc}(G).$$

Proof. Let $V^+ = \{i : (Lw)_i > 0\}$ and $V^- = \{i : (Lw)_i < 0\}$. Write $Lw = \sum_{j\in V^+, i\in V^-} \beta_{ij}(e_j - e_i)$, with $\beta_{ij} \geq 0$. This is clearly possible since $\sum_i (Lw)_i = 0$. Let $w^{(ij)} \in \mathbb{R}^V$ be the unique solution of

$$Lw^{(ij)} = e_j - e_i, \qquad w^{(ij)} \geq 0, \qquad , w_j^{(ij)} = 0,$$

and let $w' = \sum_{j\in V^+, i\in V^-} \beta_{ij} w^{(ij)}$. Then $L(w' - w) = 0$, whence $w' - w = \alpha v$ for some strictly positive vector v, by Proposition 3.1. So, $w' - w$ is either all-positive, or all-negative, or zero. Since $\min_k w_k = 0$ and $w' \geq 0$, this implies that $w \leq w'$. Hence

$$\|w\| \leq \|w'\| = \sum_{i,j} \beta_{ij} \|w^{(ij)}\| = \sum_{i,j} \beta_{ij} \text{ acc}(i,j)$$
$$\leq \left(\sum_{i,j} \beta_{ij}\right) \text{ acc}(G) = \frac{1}{2}|Lw| \text{ acc}(G).$$

∎

4. Period length and game length

Let $G = (V, E)$ be a digraph with Laplacian L. A vector $v \in \mathbb{R}^V$ is called a *period vector* for G if it is non-negative, integral, and $Lv = 0$. The name comes from the observation that if v is a period vector in a chip-firing game on G, and every node i is fired v_i times, then the beginning and ending positions are the same. Such a game can therefore be repeated any number of times. A period vector is *primitive* if its entries have no non-trivial common divisor.

The discussion in connection with Proposition 3.1 implies the following.

4.1 Proposition. (i) *Every strongly connected digraph G has a unique primitive period vector v_G. It is strictly positive, and all period vectors are of the form $tv_G, t = 1, 2, \ldots$.*

(ii) *If G is connected eulerian, then $v_G = \mathbf{1}$.*

(iii) *In general, the period vectors of a digraph are exactly the vectors of the form $\sum_{i=1}^k \lambda_i v_i, \lambda_i \in \mathbb{Z}_+$, where $v_1, \ldots, v_k$ are the primitive period vectors of the sink components.*

We call $|v_G|$ the *period length* $\text{per}(G)$ of the strongly connected digraph G. We extend this definition to all digraphs by defining $\text{per}(G)$ as the sum of $\text{per}(H)$ over all strongly connected components H of G. For instance, if all strongly connected components of G are eulerian then $\text{per}(G) = n$.

It follows from Lemma 3.2 that for the stationary distribution q on a strongly connected digraph G and $v = v_G$:

$$q_i = \frac{d^+(i)v_i}{\|v\|}. \tag{4.1}$$

Corollary 3.4 implies that

$$\mathrm{acc}(G) \le \sum_k \frac{d^+(k)}{q_k} = \|v\| \sum_k 1/v_k \le n\|v\|, \tag{4.2}$$

and hence:

4.2 Proposition. *For every strongly connected digraph,*

$$\mathrm{acc}(G) \le nD\ \mathrm{per}(G).$$

∎

(We suspect that the quotient nD we loose here is far too generous.)

Our aim in this section is to relate the period length of G to the maximal length of a finite game playable on G. For this we will first derive a crucial combinatorial property of the game language, which will also be important for the proof of Theorem 5.1.

4.3 Lemma. *Let v be a period vector of the digraph G, and let α be a legal chip-firing game (from some initial distribution). Let α' be the subword obtained by deleting the first v_i occurences of node i in α for all i (if i occurs fewer then v_i times, we delete all of its occurences). Then α' is a legal game.*

Proof. Let $\alpha = x_1 \dots x_m$ and $\alpha' = x_{i_1} \dots x_{i_k}$. We may assume (by induction) that $x_{i_1} \dots x_{i_{k-1}}$ is legal. Let $y = x_{i_k}$. In game α, y has $c \ge d^+(y)$ chips on it after the first $i_k - 1$ moves; let us see how this number compares with the number of chips on node y after $k-1$ moves in game α'. We have deleted $v(y)$ occurences of y before the current one, so y was fired $v(y)$ fewer times, which leaves $(d^+(y) - d_{y,y})v(y)$ more chips on y. But its neighbors have also been fired less often, so y receives fewer chips from them. More exactly, from each node u, if y received $a(u)$ chips from u after $i_k - 1$ moves in the game α then it receives $\max\{0, a(u) - d_{u,y}v(u)\}$ chips after $k-1$ moves in the game α'. This is a loss of $\min\{d_{u,y}v(u), a(u)\}$. Hence the number of chips on y after the first $k-1$ moves in game α' is

$$c+(d^+(y)-d_{y,y})v(y)-\sum_{u:u\ne y} \min\{d_{u,y}v(u), a(u)\} \ge c+d^+(y)v(y)-\sum_u d_{u,y}v(u) = c \ge d^+(y)$$

(by the definition of a period vector). Hence α' is legal. ∎

Let us call a non-negative vector $a \in \mathbb{R}^V$ *reduced*, if for every period vector v, there exists a node i with $a_i < v_i$.

4.4 Proposition. *Every score vector of a finite game is reduced.*

Proof. Assume α is a legal game with non-reduced score vector a. Then Lemma 4.3 gives a legal game α' with score vector $a - v$ for some period vector v. Now, by Proposition 1.2, α' can be augmented from α to a legal game $\alpha'\beta$ with the same score as α. So α' and $\alpha'\beta$

lead to the same position (since $[\beta] = v$ is a period vector), and hence $\alpha'\beta\beta\ldots$ is a legal infinite game. Also α and $\alpha'\beta$ lead to the same position (having the same score), so also $\alpha\beta\beta\ldots$ is a legal infinite game. ∎

Since eulerian graphs have period vector $\mathbf{1}$, the following generalization of Tardos' result for undirected graphs can be deduced.

4.5 Corollary. *If a game on an eulerian graph is finite then there is a node that is never fired.*

For every strongly connected digraph G, we have introduced two parameters: the access time $\mathrm{acc}(G)$ and the period length $\mathrm{per}(G)$. Both in some sense measure the speed of a certain "diffusion" process. A further such parameter is the *game length* $\mathrm{game}(G)$, the maximum length of any finite game on the graph. For an undirected graph, the period length is n, the access time is $O(n^3)$, while (by Tardos's theorem) the game length is $O(n^4)$.

We are going to show that the game length exceeds the period length by at most a polynomial factor, thereby extending Tardos's theorem to directed graphs (up to the degree of the polynomial). We conjecture that a reverse such inequality also holds.

4.6 Lemma. *Let G be a strongly connected digraph with primitive period vector v, and let a be a reduced non-negative vector. Then*

$$\|a\| \le (1 + \frac{n}{2}|La|)\|v\|.$$

Proof. By the definition of reducedness, there exists a number $0 \le \lambda < 1$ such that $\min_k a_k - \lambda v_k = 0$. Hence by Proposition 3.6,

$$\|a - \lambda v\| \le \frac{1}{2}|L(a - \lambda v)|\ \mathrm{acc}(G).$$

Now here $L(a - \lambda v) = La$, by the definition of the period vector, so using (4.2) we get

$$\|a\| \le \lambda\|v\| + \frac{1}{2}|La|\ \mathrm{acc}(G) \le \|v\| + \frac{1}{2}|La|n\|v\|.$$

∎

4.7 Lemma. *Let a be the score vector of a game with N chips on a digraph G. If a is reduced, then $|a| < 2nND\ \mathrm{per}(G)$.*

Proof. Let p and q be the initial and final positions of the game. Then $|La| = |q-p| \le 2N$, since $|p| = |q| = N$. By Lemma 4.6, if G is strongly connected then

$$|a| \le \|a\| \le (1 + nN)\|v\| < 2nND\ \mathrm{per}(G). \qquad (4.3)$$

We will derive a similar bound for all digraphs. For each strongly connected component H of G, let a_H be the restriction of the vector a to the positions in H. So $|a_H|$ is the number of firings in that component. We may assume that if a component H_1 can be reached from a component H_2 on a directed path, then we execute all firings in H_2 before firing anything in H_1: firing a node of H_1 never helps in H_2.

If H is a sink component, then

$$|a_H| < 2nND \text{ per}(H) \tag{4.4}$$

follows from (4.3). If H is not a sink component, then H has a non-empty set U of nodes which are connected to some node outside H. Now, any time a node in U is fired, at least one chip is lost from H forever, so nodes in U can be fired at most N times. All firings in H form a game in H (the chips sent out from a node $u \in U$ are considered to be left on u). Hence we can estimate $|a_H|$ similarly as in Lemma 4.6: let v_H be the primitive period vector of H, and $\lambda \geq 0$ such that $\min_{k\in H}(a_k - \lambda(v_H)_k) = 0$. Then by our observation about the elements of U, we have $\lambda \leq N$. By Propositions 3.6 and 4.2,

$$\|a_H - \lambda v_H\| \leq \frac{1}{2}|L_H(a_H - \lambda v_H)| \text{ acc}(H) = \frac{1}{2}|L_H(a_H)| \text{ acc}(H) \leq NnD \text{ per}(H),$$

and hence

$$|a_H| \leq \|a_H\| \leq \|a_H - \lambda v_H\| + \lambda\|v_H\| \leq nND \text{ per}(H) + ND \text{ per}(H) < 2nND \text{ per}(H). \tag{4.5}$$

Summing (4.4) or (4.5) over all strongly connected components, we obtain (4.3) for an arbitrary digraph. ∎

4.8 Theorem. *For every directed graph,*

$$\text{game}(G) \leq 2nmD \text{ per}(G).$$

Proof. Let us consider a finite game of maximal length and with score vector a. By Proposition 4.4 the vector a is reduced, and it was noted in Section 2 that $N \leq m - n$. Hence Lemma 4.7 gives: $\text{game}(G) = |a| < 2n(m-n)D \text{ per}(G)$. ∎

4.9 Corollary. *If every strongly connected component of G is eulerian then*

$$\text{game}(G) \leq 2n^2mD,$$

and if furthermore G has no multiple edges then

$$\text{game}(G) \leq 2n^5.$$

∎

We have seen two basic relations between the three "diffusion" parameters we considered:

$$\mathrm{acc}(G) \le nD \ \mathrm{per}(G) \text{ and } \ \mathrm{game}(G) \le 2nmD \ \mathrm{per}(G).$$

Is there any other relation of this nature?

The access time can be much smaller than the other two quantities. Consider a 2-connected undirected graph G and orient one edge (leave the rest two-way). Then one can argue that the access time remains polynomial; on the other hand, the example of Eriksson [10] is of this type (see Figure 1), and for it the game length and period length are exponentially large.

We do not know if the game length can ever be substantially smaller than the period length. As mentioned, $\mathrm{per}(G)$ can be exponentially large. However, the following bound limits the size of $\mathrm{per}(G)$, and hence also of the other two parameters.

4.10 Proposition. *For every digraph,*

$$\mathrm{per}(G) < (2D)^{n-1}.$$

Proof. Assume first that G is strongly connected. Since $\mathrm{rank} L = n-1$ we can find a non-zero minor of size $(n-1) \times (n-1)$, say the minor $L^{i,j}$ obtained by deleting row i and column j from L. The vector $u = (u_1, \dots, u_n)$ defined by $u_k = (-1)^{i+k} L^{i,k}$ is integral, non-zero, and satisfies $Lu = 0$, and is therefore (up to sign) a period vector. Furthermore, by Hadamard's inequality ($|\det(A)|$ is at most the product of the lengths of the column vectors of A) each component of u has magnitude less than $(\sqrt{2}D)^{n-1}$. Hence, per $(G) < n(\sqrt{2}D)^{n-1} < (2D)^{n-1}$.

The general case is obtained by summing the inequality over all strongly connected components of G. ∎

5. Reachability of positions

Our purpose here is to prove the following decidability result. As will be discussed at the end of the section, this is a special case of a class of decision problems whose decidability is known from the work of Kosaraju [14] and Mayr [15]. However, the nature of our algorithm and the complexity bounds obtained are new.

5.1 Theorem. *Given two positions p and q of N chips on a directed graph G, it is decidable in $O(n^2 D^2 \ \mathrm{per}(G) \log[nND \ \mathrm{per}(G)])$ time whether q can be reached from p via a sequence of chip firings.*

To describe an instance of the problem we need $n^2 \log(D+1) + 2n \log(N+1)$ bits, where the first term describes the digraph G and the second term the two positions p and q. So except for the factors of D^2 and $\mathrm{per}(G)$, the running time is polynomial in the input length. Note that, by Proposition 4.10, $\log(D \ \mathrm{per}(G)) \le n \log(2D)$.

5.2 Lemma. *Suppose that some chip-firing game leads from position p to q. Then among the score vectors of such games there is a unique one which is reduced.*

Proof. By Lemma 4.3, if there is any legal game leading from p to q, then there is one for which the score vector a is reduced. This vector a satisfies

$$La = q - p, \tag{5.1}$$

as does the score vector of any game leading from p to q.

We show that this reduced score vector is unique. For, let a_1 and a_2 be two reduced non-negative integer solutions of (5.1). Then $L(a_1 - a_2) = 0$, so by Proposition 3.1 we can write $a_1 - a_2 = \sum_i \lambda_i v_i$, where the v_i are primitive period vectors for the sink components. By reducedness, we have $|\lambda_i| < 1$; but the λ_i are integers since $a_1 - a_2$ is integral and v_i is primitive. Hence $\lambda_i = 0$ for all i, i.e., $a_1 = a_2$. ∎

The following is a consequence of Lemmas 4.7 and 5.2.

5.3 Lemma. *If some chip-firing game leads from position p to q, then q can be reached from p with fewer than $2nND$ per(G) firings.*

Proof of Theorem 5.1. We can determine the reduced score vector a of a (possible) game leading from p to q by solving (5.1) and also computing the period vectors of G. By subtracting or adding appropriate multiples of them to a, we may assume that a is reduced. If no reduced a is obtained this way we conclude that q is not reachable from p.

We want to decide whether a is the score vector of a legal game with beginning position p, since this is the case if and only if q is reachable from p. Let $\mathcal{L}$ denote the language consisting of all legal games from beginning position p, and $\mathcal{L}[a]$ the sublanguage consisting of all words in $\mathcal{L}$ that use letter i at most a_i times. By Lemma 1.4, $\mathcal{L}[a]$ is also a locally free permutable hereditary language, which implies that its maximal words can be found "greedily", i.e. without backtracking.

Now $\mathcal{L}[a]$ has rank at most $|a|$ and it has rank $|a|$ if and only if a itself is the score vector of a legal game beginning at p. So finding any maximal word in $\mathcal{L}[a]$ tells us whether a is the score of any legal game starting at p. Such a word can be found in at most $|a|$ steps by successive firing of legal vertices.

We will be a bit careful, however, in finding this word since our upper bound on $|a|$ contains the factor N, the number of chips, and this may be arbitrarily large. Note, however, that if N is large then we can carry out many firings simultaneously. Let, at any stage of the game, $\hat{a}_i$ denote the number of times node i has to be fired later on (equivalently, $a_i - \hat{a}_i$ is the number of times i has been fired). Let N_i denote the number of chips on node i at that stage. We call a node with $\hat{a}_i = 0$ *frozen*, and we denote by $\widehat{N}$ the number of chips on unfrozen nodes. We can take off the chips from the frozen nodes and get, by Lemma 4.7 that the remaining number of firings is

$$|\hat{a}| < 2nD\widehat{N} \text{ per}(G).$$

Now there is an unfrozen node, say node i, containing at least $\widehat{N}/n$ chips. If $N_i \geq \hat{a}_i d^+(i)$, then we can fire i simultaneously $\hat{a}_i$ times, and thereby freeze it. Otherwise, we can fire node i simultaneously $[N_i/d^+(i)]$ times, and reduce $|\hat{a}|$, the remaining number of firings, by at least

$$\frac{N_i}{d^+(i)} \geq \frac{\widehat{N}}{nD} > \frac{|\hat{a}|}{2n^2D^2 \ \mathrm{per}(G)}.$$

So the remaining number of firings is reduced by a factor of $1 - 1/(2n^2D^2 \ \mathrm{per}(G))$.

Now, the number of steps in which a new node is frozen is at most n. If there are $t+1$ steps of the other kind, then

$$|a| \left(1 - \frac{1}{2n^2D^2 \ \mathrm{per}(G)}\right)^t \geq 1,$$

which implies that

$$t < 2n^2D^2 \ \mathrm{per}(G) \log|a| < 2n^2D^2 \ \mathrm{per}(G) \log(2nND \ \mathrm{per}(G)).$$

Since this expression dominates the number of steps needed for computing the reduced score vector a, which is $O(n^3)$, the proof is complete. ∎

5.4 Corollary. *Given two positions p and q of chips on an eulerian directed graph without multiple edges (e.g. a simple undirected graph), it is polynomial time decidable whether q can be reached from p.*

We suspect that for a general digraph, the reachability problem is NP-hard (or perhaps even harder: we don't see that it would be either in NP or co-NP).

One can, of course, state the reachability problem for vector addition systems: given a vector addition language $\mathcal{L}$, as defined in Section 1, and a vector $a \in C$, is there a sequence $v_1 \dots v_k \in \mathcal{L}$ such that $b + v_1 + \cdots + v_k = a$? This is a well-known problem, which was shown by Kosaraju [14] and Mayr [15] (see also [18, Ch. 5]) to be decidable for every integral vector addition system. The computational complexity of such decision procedures was left open by them. Our results can be extended to vector addition systems with property $(*)$; we omit the details.

6. The "Probabilistic Abacus"

Consider a directed graph G with h sink nodes $t_1, \dots, t_h$, and a further specified non-sink node s. Let us assume that there are no other sink components. We start a random walk at s; this terminates when it reaches one of $t_1, \dots, t_h$. We are interested in the probability p_i that it terminates at sink t_i, and in the access time $\mathrm{acc}(s, t_i)$. The probability p_i can, of course, be computed using the theory of Markov chains and basic linear algebra; and an elaborate theory exists for computing the access (or passage) time, see Kemeny and Snell [12] or Kemperman [13]. We now want to discuss a combinatorial procedure for these tasks, which exhibits a further connection between chip firing and random walks.

The "probabilistic abacus", due to A. Engel [7,8], is a chip-firing procedure on G, in which the terminal nodes $t_1, \dots, t_h$ are never allowed to be fired (this can be achieved in our usual game model by attaching sufficiently many loops to them). We start with placing $d^+(k) - 1$ chips on each non-sink node k. We call this the "critical position"; no node can be fired. Now place an additional chip on s and play the chip firing game until it terminates. Call this the first phase. If it terminates with the critical position on the non-sink nodes, stop; otherwise, feed a new chip to s and play the game until it terminates; call this the second phase, etc. We go on like this as long as the critical position does not reappear on the non-sink nodes; if it does then we stop.

6.1 Theorem. (i) The probabilistic abacus terminates after at most D^{n-h} phases.
(ii) The number of phases can grow exponentially with n.

Assume that we needed M phases, and that a is the score vector of a complete run of the probabilistic abacus. Let M_j be the number of chips on sink t_j at termination. Clearly, $\sum_{j=1}^{h} M_j = M$.

6.2 Theorem. (Engel [7,8]) (i) $p_j = M_j/M$, for $j = 1, \dots, h$.
(ii) If $h = 1$, then $\operatorname{acc}(s, t_1) = \|a\|/M$.

Although the finite termination of the probabilistic abacus is necessary for its performance, there seems to be no previously available proof for this property. As we will show, it follows rather easily from Theorem 1.1. Engel [8] writes: "We have proved all our claims with one exception: We have not shown that the critical load always recurs. For this conjecture we have only the strong evidence of about 1000 examples." At the end of the same paper there is a "Note added in proof" announcing that L. Scheller has found a proof of the critical load conjecture. The note mentions a main idea of Scheller's proof (the "freezing" of chips) which is also one of the key ingredients of our proof below. A proof was also independently found by Rimányi [19], at about the same time as ours. We have not been able to find a complete proof published by L. Scheller or by anyone else.

While one run of the probabilistic abacus simultaneously computes the absorbtion probabilities at all the sinks $t_1, \dots, t_h$ by part (i) of Theorem 6.2, part (ii) shows that a separate run is needed for each sink (with the others deleted) to compute the access times. One way to speed up the performance of the probabilistic abacus is to use simultaneous firings, as we did in the proof of Theorem 5.1 (if node k has $d^+(k)q+r$ chips and $r < d^+(k)$ then move q chips at once along each edge leaving k). This is actually the firing rule given by Engel. While this speed-up will somewhat shorten each phase, it does not affect the total number of phases. One can also ask how long an individul phase of the abacus can be. For this we point to the general bound given by Theorem 4.8. It follows from Theorems 4.8, 4.10 and 6.1 that a complete run of the probabilistic abacus has at most $2nmD \operatorname{per}(G) D^{n-h} \le n^2(2D^2)^n$ firings.

Proof of Theorem 6.1 (i). The existence of sinks $t_1, \dots, t_h$ that may not be fired, and the absence of other sink components, shows by Lemma 2.1 that every chip game on G is finite. In particular, every phase of the probabilistic abacus is finite. The terminating position of each phase has at most $d^+(k) - 1$ chips on each non-sink node k, so there are at most D^{n-h} such positions. Therefore, after at most D^{n-h} phases some terminating position q on the non-sink nodes must reappear. Suppose that it takes M phases to play from the first appearance of q to the second. (Note that by Theorem 1.1 the sequence of these terminating positions is predetermined and independent of how each phase is played.)

We now construct a new graph G' from G by creating a new node r and joining it to s by a directed edge. Place M chips on r and $d^+(k)-1$ chips on each "old" non-sink node k. Then the first M phases of the abacus can be mimicked as a single chip game on G', governed by the rule that r is fired only when necessary. Call this Game 1.

But we can also play a chip game on G' from the given position as follows: we "freeze" $d^+(k)-1-q(k)$ of the chips on each "old" node k, and then mimic with the remaining chips the M phases that lead from the first appearance of the position q to the second. Call this Game 2. By construction, Game 2 terminates with the critical position on the non-sink nodes and with no chips on node r. Therefore by Theorem 1.1 the same is true about Game 1. But this means that the critical position will reappear after M phases of the probabilistic abacus. ∎

Proof of Theorem 6.2. For part (i) we want to view the whole run of the probabilistic abacus as one period of a single chip firing game. For this, enlarge G to a new graph G'' by creating a new node r, connect r to s by a directed edge and connect each t_j to r by a directed edge. The initial position on G'' will be the critical position on the "old" non-sink nodes and one single chip on r. If we play according to the rule: (i) only fire r if we must, (ii) fire each t_j whenever we can, then we will exactly imitate the probabilistic abacus. When after M phases the critical position returns, we have completed one period of the game on G''. Furthermore, the number of times t_j was fired is the same as the number M_j of chips accumulating on t_j in the probabilistic abacus. So the numbers M_j are the t_j-entries of a period vector for the strongly connected digraph G''.

It is easy to relate random walks on G to random walks on G'': whenever a random walk on G terminates, it goes on in G'' by returning to r and then to s, and starting again. Hence (by doing this for very long) we see that the p_j are proportional to the probabilities of the nodes t_j in the stationary distribution of the random walk on G''. By (4.1) these are proportional to the corresponding entries of any period vector, and hence to the M_j. This proves part (i).

In the situation of part (ii) there is only one sink $t = t_1$. In addition to the critical position, place M extra chips on node s. Then the whole run of the probabilistic abacus can be seen as a single game on G with score vector a. This game results in the net transport of the M extra chips from s to t, so $La = M(e_t - e_s)$. Therefore by Lemma 3.5: $\mathrm{acc}(s,t) = \|a\|/M$. (For this lemma, G is supposed to be strongly connected. This can be arranged by adding a directed edge from t to s, which is otherwise harmless.) ∎

Proof of Theorem 6.1 (ii). Consider the digraph G in Figure 1 with $n+1$ nodes, n even. Removal of the sink t leaves a subgraph G', which has been studied by Eriksson [10]. Eriksson showed that an exponentially long chips game can be played on G', and the idea here is to use his analysis to show that the probabilistic abacus on G has exponentially many phases.

Figure 1. The graph G, $|V| = n + 1$.

All edges of G are bidirected, except for the edges su_0 and st. Chips distributions on G will be denoted by sequences of the type $(x_s; \dots, x_{u_{-2}}, x_{u_{-1}}, x_{u_0}, x_{u_1}, x_{u_2}, \dots; x_t)$, where x_k is the number of chips on node k. Note that $d^+(s) = n$, $d^+(u_0) = 2$, and $d^+(u_i) = 3$ for all $i \neq 0$.

We will now describe a few moves of the first phase of the probabilistic abacus on G:

Start: critical position	$(n-1; \dots, 2, 2, 1, 2, 2, \dots; 0)$
Add one chip to s and fire	$(0; \dots, 3, 3, 2, 3, 3, \dots; 1)$
Fire each u_i once	$(n-2; \dots, 2, 2, 2, 2, 2, \dots; 1)$
Fire the u_i's clockwise, beginning and ending with u_0	$(2n-4; \dots, 1, 2, 0, 2, 1, \dots; 1)$
Fire s	$(n-4; \dots, 2, 3, 1, 3, 2, \dots; 2)$
Fire the u_i's clockwise, beginning with u_1 and ending with u_0	$(2n-6; \dots, 1, 2, 1, 2, 1, \dots; 2)$

The position on the subgraph G' at this stage is called P_1 by Eriksson [10]. From here on we continue playing in the manner described by him, with the additional rule that whenever s should be fired but has too few chips (due to the loss of chips to t) then new chips may be added to s to make its firing legal. Of course, each time a new chip is added a new phase of the probabilistic abacus on G is started. Eriksson shows that this procedure will end with the critical position on G'. Adding one more chip to s (one more phase) we get back the critical position on the non-sink nodes of G, and Eriksson's construction is such that it cannot have reappeared before.

The number of phases of the complete run of the abacus just described is equal to the number of new chips fed to s, which is equal to the number of chips on the sink t at termination, which is in turn equal to the number of times node s was fired. The number of firings of s from P_1 to termination in the abacus on G is by construction equal to the number of firings of s from P_1 to termination in Eriksson's game on G'. In both cases there are 2 additional firings (to reach P_1), so the number of firings of s can be computed for Eriksson's game on G' instead of for the abacus on G. This will be done with a method that parallels Eriksson's computation of the *total* number of firings on G'.

Let P_k be the position on G' with 2 chips on nodes u_i for $i = \pm 1, \pm 2, \ldots, \pm k$, one chip on all other u_i's and $2n-4-2k$ chips on s. Eriksson describes how to play from P_{k-1} to P_k, $k = 1, 2, \ldots$, until the final (critical) position $P_{(n-2)/2}$ is reached. If b_k is the number of firings of s when going from P_{k-1} to P_k, then $b_1 = 2$ and $b_k = 2b_{k-1} + \sum_{i=1}^{k-2} b_i, k \geq 2$. Hence, the total number of firings of s when going from P_0 to P_k, call it $c_k = b_1 + \cdots + b_k$, satisfies the linear recursion

$$c_k - 3c_{k-1} + c_{k-2} = 0\ ,\ \ c_0 = 0, c_1 = 2,$$

which has solution

$$c_k = \frac{2}{\sqrt{5}} \left(\tau^{2k} - \tau^{-2k}\right)\ ,\ \ k \geq 0,$$

where $\tau = (1+\sqrt{5})/2$. Therefore, the number of times s is fired when going from P_0 to $P_{(n-2)/2}$, which equals the number of phases of the probabilistic abacus on G, is

$$c_{(n-2)/2} = \frac{2}{\sqrt{5}} \left(\tau^{n-2} - \tau^{2-n}\right) \sim \frac{2}{\sqrt{5}} \tau^{n-2}.$$

$\blacksquare$

References

1. R. Aleliunas, R. M. Karp, R. J. Lipton, L. Lovász, and C. W. Rackoff, Random walks, universal travelling sequences, and the complexity of maze problem, Proc. 20th Ann. Symp. on Foundations of Computer Science, pp. 218–223, 1979.
2. N. Alon, Eigenvalues and expanders, *Combinatorica* **6** (1986), 83–96.
3. R. J. Anderson, L. Lovász, P. W. Shor, J. Spencer, É. Tardos, and S. Winograd, Disks, balls, and walls: analysis of a combinatorial game, *Amer. Math. Monthly* **96** (1989), 481-493.
4. A. Björner, L. Lovász, and P. W. Shor, Chip-firing games on graphs. *European J. Comb.* **12** (1991), 283–291.
5. A. Björner and G. M. Ziegler, Introduction to greedoids, in: *Matroid Applications* (N. White, ed.), Cambridge Univ. Press, Cambridge, pp. 284–357, 1992.
6. P. Diaconis and W. Fulton, A growth model, a game, an algebra, Lagrange inversion, and characteristic classes, preprint, 1991.
7. A. Engel, The probabilistic abacus, *Educ. Stud. in Math.* **6** (1975), 1–22.
8. A. Engel, Why does the probabilistic abacus work? *Educ. Stud. in Math.* **7** (1976), 59–69.
9. K. Eriksson, A game of chips and cards, preprint, 1990.
10. K. Eriksson, No polynomial bound for the chip-firing game on directed graphs, *Proc. Amer. Math. Soc.* **112** (1991), 1203–1205.
11. R. M. Karp and R. E. Miller, Parallel program schemata, *J. Computer & System Sci.* **3** (1969), 147–195.
12. J. G. Kemeny and J. L. Snell, *Finite Markov Chains* , Van Nostrand, Princeton, NJ, 1960.
13. J. H. B. Kemperman, *The passage problem for a stationary Markov chain*, Univ. of Chicago Press, Chicago, 1961.
14. S. R. Kosaraju, Decidability of reachability in vector addition systems, Proc. 14th Ann. ACM Symp. on Theory of Computing, pp. 267–281, 1982.
15. E. W. Mayr, An algorithm for the general Petri net reachability problem, *SIAM J. Comput.* **13** (1984), 441–460.
16. H. Minc, *Nonnegative matrices*, J. Wiley & Sons, New York, 1988.
17. W. Reisig, *Petri Nets: An Introduction*, Springer-Verlag, New York, 1985.
18. C. Reutenauer, *Aspects mathématiques des réseaux de Petri*, Masson, Paris, 1988.
19. R. Rimányi, unpublished manuscript, 1988.
20. J. Spencer, Balancing vectors in the max norm, *Combinatorica* **6** (1986), 55–66.
21. G. Tardos, Polynomial bound for a chip firing game on graphs, *SIAM J. Discrete Math.* **1** (1988), 397–398.