Jan van den Heuvel

Algorithmic aspects of a chip-firing game

Article (Published version) (Refereed)

Original citation:
van den Heuvel, Jan (2001) Algorithmic aspects of a chip-firing game. Combinatorics, probability & computing, 10 (6). pp. 505-529. ISSN 0963-5483
DOI:10.1017/S0963548301004886

Combinatorics, Probability and Computing (2001) 10, 505–529.
DOI: 10.1017/S0963548301004886 Printed in the United Kingdom

Algorithmic Aspects of a Chip-Firing Game

JAN VAN DEN HEUVEL

Centre for Discrete and Applicable Mathematics, Department of Mathematics,
London School of Economics, Houghton Street, London WC2A 2AE, UK
(e-mail: jan@maths.lse.ac.uk)

Received 19 August 1999; revised 9 May 2001

Algorithmic aspects of a chip-firing game on a graph introduced by Biggs are studied. This variant of the chip-firing game, called the dollar game, has the properties that every starting configuration leads to a so-called critical configuration. The set of critical configurations has many interesting properties. In this paper it is proved that the number of steps needed to reach a critical configuration is polynomial in the number of edges of the graph and the number of chips in the starting configuration, but not necessarily in the size of the input. An alternative algorithm is also described and analysed.

1. Introduction

In the classical *chip-firing game* on a finite undirected graph G (see [9] and references therein) it is assumed that, at the beginning of the game, a pile of chips is placed on each vertex of G. A step in the game consists of choosing a vertex v which has at least as many chips as its degree, and moving one chip from v to each of its neighbours. Such a step is called *firing vertex* v. The game terminates if each vertex has fewer chips than its degree.

In this paper we study a variant of the classical chip-firing game above. In this variant there is one special vertex, say q, which is always able to fire, independently of its number of chips. Biggs [4, 5] calls this variant the *dollar game*, and refers to q as *the government*. In fact, the game we consider is a variant of this game, but because the differences are small we will call our variant the dollar game as well.

Before we continue, we need some definitions and will then give a formal definition of our version of the dollar game. A good source for the basic terminology and notation in graph theory is [11]. Throughout this paper we will assume as given a graph $G = (V(G), E(G))$, which is finite, undirected and connected. We also assume that there is one special vertex in G, called q. We shall denote the number of vertices of G by n and the number of edges by m. The edge-connectivity of G will be denoted by λ.

We will allow G to have multiple edges, but no loops. (Note that this means that formally $E(G)$ is a multiset. We will still talk about an edge $e \in E(G)$.) For two vertices u, v, $e(u, v)$ denotes the number of edges joining u and v; this number is also called the *multiplicity* of an edge between u and v. The *degree* $d_G(v)$ of a vertex v is the number of edges incident with v, where multiple edges are counted with their multiplicities. The set of neighbours of a vertex v is denoted by $N_G(v)$.

An instance of the *dollar game* on the graph G starts with a number of chips on each vertex $v \neq q$, where we allow the number of chips to be negative. One move in the game consists of one of the following two steps.

(C1) Choose a vertex $v \neq q$ which has more than $d_G(v)$ chips, remove $d_G(v)$ chips from that vertex, and add $e(u, v)$ chips to each vertex $u \in N_G(v) \setminus \{q\}$. Such a step is called *firing vertex v*.

(C2) If there is no vertex $v \neq q$ that has more than $d_G(v)$ chips, then add $e(u, q)$ chips to each vertex $u \in N_G(q)$. This step is called *firing q* or *firing the government*.

The number of chips on a vertex $v \neq q$ at a certain moment will be denoted by $s(v)$. A *configuration* $\boldsymbol{s}$ of the dollar game is the $(n-1)$-vector of all numbers of chips at a certain moment in the game. If $s(v) \geqslant d_G(v)$ for some $v \neq q$, then we say that v is *ready in $\boldsymbol{s}$*; if $s(v) < d_G(v)$ for all $v \neq q$, then q is *ready*. Given a configuration $\boldsymbol{s}$, a finite sequence $v_1, v_2, \ldots, v_k$ of vertices is *legal for $\boldsymbol{s}$* if v_1 is ready in $\boldsymbol{s}$, v_2 is ready in the configuration obtained from $\boldsymbol{s}$ after firing v_1, *etc.* If $v_1, \ldots, v_k$ is a legal sequence for $\boldsymbol{s}$, then by *applying the sequence* we will mean the process of firing the vertices of the sequence in consecutive order, starting with the configuration $\boldsymbol{s}$.

A configuration $\boldsymbol{s}$ is said to be *stable* if $s(v) < d_G(v)$ for all $v \neq q$; it is called *recurrent* if there is a non-empty legal sequence for $\boldsymbol{s}$ which leads to the same configuration. Further, a configuration is *critical* if it is both stable and recurrent.

Although our version of the dollar game differs from that in [4] in that we allow a negative number of chips, most of the theory developed in [4] holds without any changes. In particular, the following results, which are the main inspiration for the results in this paper, are still valid.

Theorem 1.1 (Biggs [4]). *Let $\boldsymbol{s}$ be a configuration of the dollar game on a connected graph G. Then there exists a unique critical configuration $\boldsymbol{c}$ which can be reached by a legal sequence of firings starting from $\boldsymbol{s}$.*

Denote the set of critical configurations of G by K, the set of spanning trees of G by $\mathcal{T}$, and the number of spanning trees by κ.

Theorem 1.2 (Biggs [4]). *Let G be a connected graph. Then the number of elements in K is equal to κ, the number of spanning trees of G.*

For each configuration $\boldsymbol{s}$, let $\gamma(\boldsymbol{s}) \in K$ be the unique critical configuration determined by the previous theorem. For two critical configurations $\boldsymbol{c}_1, \boldsymbol{c}_2 \in K$, define the operation '$\bullet$' by $\boldsymbol{c}_1 \bullet \boldsymbol{c}_2 = \gamma(\boldsymbol{c}_1 + \boldsymbol{c}_2)$ (where '$+$' is the normal vector addition).

Theorem 1.3 (Biggs [4]). *Let G be a connected graph. Then the set K with the operation $\bullet$ is an abelian group.*

It is possible to say much more about the relations between the set of critical configurations and other properties of the graph G: see, for instance, [4, 5, 6, 19]. In that sense it seems a natural question to ask about algorithmic aspects of these results. In particular the following questions seem to be of interest.

(Q1) Given a configuration $\boldsymbol{s}$, how long does it take to compute $\gamma(\boldsymbol{s})$?
(Q2) Given two critical configurations $\boldsymbol{c}_1, \boldsymbol{c}_2$, how long does it take to compute $\boldsymbol{c}_1 \bullet \boldsymbol{c}_2$?
(Q3) Does there exist a bijection from K and $\mathscr{T}$ that is efficiently computable?

Question (Q3) is essentially answered in the affirmative in [7]. An alternative description will be given in the sequel paper [17], the description in which makes it easier to determine the following result. As units of complexity we will use a *chip movement*, which is the operation of moving one chip from one vertex to a neighbour, and a *firing*, which is the operation of firing one vertex. Notice that firing a vertex v involves $d_G(v)$ chip movements.

Theorem 1.4 (Biggs and Winkler [7]). *For a connected graph G, there exist a bijection $f : K \to \mathscr{T}$, such that, for any critical configuration $\boldsymbol{c} \in K$, determining $f(\boldsymbol{c})$ involves $O(n)$ firings, or $O(m)$ chip movements.*

Answers to questions (Q1) and (Q2) will follow from results in Section 7, where the following results are obtained. Define the norm $\|\cdot\|$ of a configuration $\boldsymbol{s}$ by

$$\|\boldsymbol{s}\| = \sum_{v \in V(G) \setminus \{q\}} |s(v)|.$$

Theorem 1.5. *For a connected graph G and a configuration $\boldsymbol{s}$ on G, the critical configuration $\gamma(\boldsymbol{s})$ can be determined in $O(n^2\,(\|\boldsymbol{s}\| + m)\,\lambda^{-1})$ firings, involving $O(n\,m\,(\|\boldsymbol{s}\| + m)\,\lambda^{-1})$ chip movements.*

Theorem 1.6. *For a connected graph G and two critical configurations $\boldsymbol{c}_1, \boldsymbol{c}_2$ on G, the sum $\boldsymbol{c}_1 \bullet \boldsymbol{c}_2$ can be determined in $O(n^2\,m\,\lambda^{-1})$ firings, involving $O(n\,m^2\,\lambda^{-1})$ chip movements.*

Note that the bounds on the number of chip movements in Theorems 1.5 and 1.6 are polynomial in n and m, but not necessarily in the *size of the input* (see below). In particular, if $\|\boldsymbol{s}\|$ is large, then the number of movements in Theorem 1.5 will be large as well. This is why a different kind of procedure is described and analysed in Section 8. This procedure will not only involve chip movements, but also certain elementary arithmetic calculations (addition, multiplication) of rational numbers with numerator and denominator of the order $\max\{\kappa, \max_{v \in V(G) \setminus \{q\}} |s(v)|\}$.

Theorem 1.7. *For a connected graph G and a configuration $\boldsymbol{s}$ on G, the critical configuration $\gamma(\boldsymbol{s})$ can be determined in a procedure involving $O(n^2)$ arithmetic operations and $O(n^2\,m\,\lambda^{-1})$ firings, involving $O(n\,m^2\,\lambda^{-1})$ chip movements.*

In order to decide if the previous results give algorithms that are polynomial in the size of the input, we have to tell what we mean by the size of the input. We follow the more or less standard convention (see, *e.g.*, [16]) and define it as the length of a binary encoding of an instance of the problem. Encoding a nonnegative integer x takes $\lceil \log_2(x+1) \rceil$ bits, whereas encoding an integer z takes $\lceil \log_2(|z|+1) \rceil + 1$ bits (one extra bit for the sign). Since an instance of the chip-firing problem involves a graph G with possible multiple edges and an initial configuration $\boldsymbol{s}$, we need the following number of bits:

$$s(G, \boldsymbol{s}) = \sum_{u,v \in V(G),\, u \neq v} \lceil \log_2(e(u,v)+1) \rceil + \sum_{v \in V(G) \setminus \{q\}} \lceil \log_2(|s(v)|+1) \rceil + n - 1.$$

Notice that both m and $\|\boldsymbol{s}\|$ can be non-polynomial in $s(G, \boldsymbol{s})$. Hence Theorems 1.4 to 1.7 do not give polynomially bounded procedures. In the case of Theorem 1.7 we can make some more precise conclusions. The number of spanning trees of a graph G is certainly at most $\kappa \leqslant \prod_{u,v \in V(G)} e(u,v)$, so $\log_2 \kappa \leqslant \sum_{u,v \in V(G)} \log_2 e(u,v)$. Hence the size of the rationals used in the arithmetic operations from Theorem 1.7 are of the same order as the size of the input. Since performing an elementary arithmetic operation involves a number of steps polynomial in the size of the input, we obtain that the total number of operations for the arithmetic operations in Theorem 1.7 is bounded by a polynomial in the size of the input. This means that the number of operations according to Theorem 1.7 is not polynomial in the size of the input if and only if m is not polynomially bounded by the input.

In particular, we obtain that, if G is a simple graph (hence $m \leqslant \frac{1}{2} n^2$), then Theorems 1.4, 1.6 and 1.7 give a polynomial bound on the number of operations. We do not know how to bound the number of firings or the number of chip movements by an upper bound that does not involve the number of edges m of the graph. So we cannot guarantee that an algorithm to find critical configurations will stop after a number of steps polynomial in the input if the graph has edges of very high multiplicity.

Similar results to those above have been found by others. Complexity results for the classical chip-firing game can be found in [9, 18, 22]. These results can be used to estimate the complexity of obtaining a stable configuration, and compare favourably with Corollary 7.4(a). Results on the complexity of chip-firing games on directed graphs appear in [8, 15]. It appears that on directed graphs, and even in mixed graphs in which all edges except one are undirected, it can take an exponential number of firings before a stable configuration is obtained.

The dollar game is equivalent to what is known in theoretical physics as *sandpile models*: see, *e.g.*, [13, 14]. A more general set-up is studied in so-called *avalanche models*, leading to the concept of *self-organized criticality* [1, 2].

There have been some results obtained from a physical point of view that are related to the work in this paper. A result in [21] means that the sum $\boldsymbol{c}_1 \bullet \boldsymbol{c}_2$ of two critical configurations on a connected simple graph can be found using $O(n^4)$ firings. Note that for a simple graph on n vertices and m edges we have $m = O(n^2)$, so Theorem 1.6 is better for large edge-connectivity λ.

In [20] sandpile models on d-dimensional lattices are discussed. These models are equivalent to dollar games on graphs consisting of finite box-shaped parts of a d-dimensional square cubic lattice, with one extra vertex (which plays the role of the government q) connected to all boundary vertices of the lattice part. It is shown in [20] that, in the case $d = 1$ and n vertices (which is the same as the dollar game on a cycle C_{n+1}), the stable configuration resulting from a given starting configuration can be found in $O(n \log(n))$ steps, more rapidly than by just performing the chip-firing procedure. The same paper also contains results indicating that it is unlikely that a faster algorithm exists for $d \geqslant 3$, but the case $d = 2$ remains an interesting open problem.

The remainder of this paper is organized as follows. In Section 2 we describe the basic theory of our chip-firing game, mostly repeating results from the literature. In Section 3 we look at some of the basic properties of critical configurations, in order to have a good start for the analysis of the procedure to find critical configurations. This analysis is continued in Section 4, where certain aspects about the dynamics of the dollar game are studied.

In order to be able to obtain quantitative results on the dynamics of the dollar game, we will compare the dollar game with some kind of 'continuous' version of the game. This continuous game seems easier to analyse: the analysis can be found in Sections 5 and 6. In Section 7 we will show that the dynamical behaviour of the dollar game and the continuous version are related, so that we can give quantitative results for the dollar game. These results will make it possible to obtain proofs for Theorems 1.5 and 1.6.

In the final section we take a different look at the problem to find a corresponding stable configuration for a given starting configuration. Our point of view in Section 8 will be quite different from that in the previous section. We look at the problem in a more algebraic sense, and discuss what that means in terms of algorithms.

In particular, our extension of the dollar game to allow negative values of chips in a configuration is only essential in the final section. Everything up to Section 8 applies if we limit ourselves to nonnegative chip numbers only.

2. Basic theory of chip-firing

We use the notation and definitions from the previous section. We will show the development of the basic theory of the dollar game, similar to the theory in [4]. We include some proofs for completeness, and because many of the arguments are used in later sections.

The following three lemmas are straightforward, and their proofs are omitted.

Lemma 2.1. *Let $\sigma = v_1, \ldots, v_k$ be a legal sequence for the configuration $\boldsymbol{s}$. Suppose also that the vertex u is ready in $\boldsymbol{s}$ and that u does not appear in the sequence σ. Then the sequence $u, v_1, \ldots, v_k$ is also legal for $\boldsymbol{s}$.*

Lemma 2.2. *Let $v_1, \ldots, v_{k-1}, v_k, v_{k+1}, \ldots, v_\ell$ be a legal sequence for the configuration $\boldsymbol{s}$. Suppose also that the vertex v_k is ready in $\boldsymbol{s}$ and that v_k does not appear in the sequence $v_1, \ldots, v_{k-1}$. Then the sequence $v_k, v_1, \ldots, v_{k-1}, v_{k+1}, \ldots, v_\ell$ is also legal for $\boldsymbol{s}$.*

Lemma 2.3. *Let $\sigma = v_1, \ldots, v_k$ be a legal sequence for the configuration $\boldsymbol{s}$ and suppose* *that q does not appear in the sequence σ. Let $\boldsymbol{s}'$ be a configuration such that $s'(v) \geqslant s(v)$ for all $v \in V(G) \setminus \{q\}$. Then σ is also a legal sequence for $\boldsymbol{s}'$.*

Let $\boldsymbol{s}$ be a configuration and suppose $\sigma = v_1, \ldots, v_k$ is a finite sequence of vertices which is legal for $\boldsymbol{s}$. Then for $v \in V(G)$ we denote the number of occurrences of the vertex v in σ by $x_\sigma(v)$. Combining all values of x_σ gives the n-vector $\boldsymbol{x}_\sigma$, the *representative vector of* σ.

The representative vector of a legal sequence σ gives a convenient way to describe the relation between the starting configuration $\boldsymbol{s}$ and the configuration $\boldsymbol{s}'$ obtained after σ has been applied. More precisely, for all $v \in V(G) \setminus \{q\}$, we have

$$s'(v) \;=\; s(v) - x_\sigma(v)\, d_G(v) + \sum_{u \in N_G(v)} x_{\sigma(u)}\, e(u, v). \tag{2.1}$$

The formula above is obvious since, whenever v itself is fired, it loses $d_G(v)$ chips, and whenever another vertex u is fired, v gains $e(u, v)$ chips.

The following result is proved in [4] using more involved counting arguments.

Lemma 2.4 (Biggs [4]). *Let $\sigma = v_1, \ldots, v_k$ and $\sigma' = v'_1, \ldots, v'_\ell$ both be legal sequences for the configuration $\boldsymbol{s}$. Then there exists a sequence $\tau = u_1, \ldots, u_j$ which is also legal for $\boldsymbol{s}$ and such that its representative vector $\boldsymbol{x}_\tau$ satisfies, for all $v \in V(G)$,*

$$x_\tau(v) \;=\; \max\{\, x_\sigma(v), x_{\sigma'}(v)\,\}.$$

Moreover, τ can be chosen such that its initial part is identical to σ.

Proof. We use induction on $k + \ell$. If $k = 0$ or $\ell = 0$, we are done by setting $\tau = \sigma'$ or $\tau = \sigma$, respectively.

If $v_1 = v'_1$, then we are done by applying the induction hypothesis on the configuration obtained after applying v_1 on $\boldsymbol{s}$, and the sequences $\overline{\sigma} = v_2, \ldots, v_k$ and $\overline{\sigma}' = v'_2, \ldots, v'_\ell$.

So we can assume $v_1 \neq v'_1$. In particular, we know that v_1 is ready in $\boldsymbol{s}$. If v_1 does not occur in σ', then by Lemma 2.1 the sequence $v_1, v'_1, \ldots, v'_\ell$ is also legal for $\boldsymbol{s}$. Now apply the induction hypothesis on the configuration obtained after applying v_1 on $\boldsymbol{s}$, and the sequences $\overline{\sigma} = v_2, \ldots, v_k$ and $\sigma' = v'_1, \ldots, v'_\ell$.

So we are left with the case that $v_1 \neq v'_1$ and v_1 occurs somewhere in σ'. Let i be the minimal index such that $v'_i = v_1$. Because v_1 is ready in $\boldsymbol{s}$, it follows from Lemma 2.1 that the sequence $v'_i, v'_1, \ldots, v'_{i-1}, v'_{i+1}, \ldots, v'_\ell$ is legal for $\boldsymbol{s}$. So now we can apply the induction hypothesis on the configuration obtained after applying v_1 on $\boldsymbol{s}$, and the sequences $\overline{\sigma} = v_2, \ldots, v_k$ and $\overline{\sigma}' = v'_1, \ldots, v'_{i-1}, v'_{i+1}, \ldots, v'_\ell$. □

Using Lemma 2.4, we can immediately prove the so-called 'confluency property' of the dollar game, which is part (a) of the following theorem. The other parts describe some special cases that are important in Section 4.

Theorem 2.5 (Biggs [4]). *Let G be a connected graph and let $\sigma = v_1, \ldots, v_k$ and $\sigma' = v'_1, \ldots, v'_\ell$ both be legal sequences for the configuration $\boldsymbol{s}_0$ on G. Suppose that applying these sequences leads to configurations $\boldsymbol{s}_1$ and $\boldsymbol{s}'_1$, respectively.*

(a) *There exists a configuration* $\boldsymbol{s}_2$ *which can be obtained from both* $\boldsymbol{s}_1$ *and* $\boldsymbol{s}'_1$ *by applying legal sequences.*
(b) *If no vertex appears more than once in both* σ *and* σ'*, then the configuration* $\boldsymbol{s}_2$ *in* (a) *can be obtained without firing a vertex more than once.*
(c) *If both* σ *and* σ' *do not contain* q*, and no vertex in* $V(G)\setminus\{q\}$ *appears more than once in both* σ *and* σ'*, then the configuration* $\boldsymbol{s}_2$ *in* (a) *can be obtained without firing* q *and such that no vertex in* $V(G)\setminus\{q\}$ *is fired more than once.*

Proof. From Lemma 2.4 it follows that we can find a legal sequence τ for $\boldsymbol{s}_0$, such that $x_\tau(v) = \max\{x_\sigma(v), x_{\sigma'}(v)\}$ for all $v \in V(G)$, and such that the initial part of τ is identical to σ. Let $\overline{\tau}$ be the subsequence of τ that appears after the part that is identical to σ. Then $\overline{\tau}$ is a legal sequence for $\boldsymbol{s}_1$. Of course, we can also find a legal sequence τ' for $\boldsymbol{s}_0$, such that $x_{\tau'}(v) = \max\{x_\sigma(v), x_{\sigma'}(v)\}$ for all $v \in V(G)$, and such that the initial part of τ' is identical to σ'. Let $\overline{\tau}'$ be the subsequence of τ' that appears after the part that is identical to σ'. Then $\overline{\tau}'$ is a legal sequence for $\boldsymbol{s}'_1$. So we obtain (a) if we can prove that the configurations obtained from applying $\overline{\tau}$ to $\boldsymbol{s}_1$ and from applying $\overline{\tau}'$ to $\boldsymbol{s}'_1$ are the same. But this follows from applying equation (2.1) to $\boldsymbol{x}_\tau$ and $\boldsymbol{x}_{\tau'}$ and observing that, for all $v \in V(G)$, we have $x_\tau(v) = \max\{x_\sigma(v), x_{\sigma'}(v)\} = x_{\tau'}(v)$.

Part (b) follows by observing that, if $x_\sigma(v) \leqslant 1$ and $x_{\sigma'}(v) \leqslant 1$, then $x_\tau(v) \leqslant 1$. And for part (c) we only need the extra observation that, if $x_\sigma(q) = x_{\sigma'}(q) = 0$, then $x_\tau(q) = 0$. □

3. Algorithmic properties of critical configurations

Recall that a critical configuration of a connected graph G is a configuration $\boldsymbol{s}$ on G that is both stable (*i.e.*, $s(v) < d_G(v)$ for all $v \in V(G) \setminus \{q\}$) and recurrent (*i.e.*, there exists a non-empty legal sequence for $\boldsymbol{s}$ which leads back to $\boldsymbol{s}$ after applying it). Most of the following results occur in [4], sometimes only implicitly and sometimes with a more algebraic proof than the ones given here.

Lemma 3.1 (Biggs [4]). *Let* $\boldsymbol{s}$ *be a stable configuration and suppose* $\sigma = v_1, \ldots, v_k$ *is a legal sequence for* $\boldsymbol{s}$ *in which* q *appears only once. Then every vertex of* G *appears at most once in* σ.

Proof. Note that, since $\boldsymbol{s}$ is stable, q must appear as the first vertex in σ. Suppose some vertices appear more than once in σ and let v be the first vertex that appears for the second time. Let v_i be this second appearance of v in σ. Then v_i must be ready after $v_1, \ldots, v_{i-1}$ has been applied to $\boldsymbol{s}$. Since v_i appears once in $v_1, \ldots, v_{i-1}$ and every other vertex appears at most once, we find that the number of chips on v_i after applying $v_1, \ldots, v_{i-1}$ is at most

$$s(v_i) - d_G(v_i) + \sum_{u \in N_G(v_i)} e(u, v_i) \ \leqslant\ s(v_i) - d_G(v_i) + d_G(v_i) \ =\ s(v_i).$$

But since $s(v_i) < d_G(v_i)$, this contradicts that v_i is ready after $v_1, \ldots, v_{i-1}$ has been applied. □

Corollary 3.2. *Let s be a stable configuration and suppose $\sigma = v_1, \dots, v_k$ is a legal sequence for s. Then every vertex appears in σ at most as often as q does.*

Proof. Partition σ into parts that start with q and where q does not appear later in the part. By applying Lemma 3.1 to each part, we are done. □

Corollary 3.3 (Biggs [4]). *Let s be a stable configuration and suppose $\sigma = v_1, \dots, v_k$ is a legal sequence for s such that after applying σ we return to the configuration s. Then every vertex of G (including q) appears the same number of times in σ.*

Proof. This can be proved using equation (2.1) and some linear algebra, as is done in [4]. The following proof is more intuitive and in line with our general approach. Suppose the result is false. Let v be a vertex that appears a minimal number of times in σ and that is adjacent to a vertex v' that appears more often in σ than v itself. (Since G is connected, such a vertex must exist if not all vertices appear equally often in σ.) By Corollary 3.2 it follows that $v \neq q$. Suppose v appears p times in σ. Then, when applying σ, v loses $p\, d_G(v)$ chips. On the other hand, v gains at least $p\, e(u,v)$ chips from each vertex $u \in N_G(v)$ and in fact at least $(p+1)\, e(v,v')$ chips from v'. This means that, after applying σ, v has at least

$$s(v) - p\, d_G(v) + \sum_{u \in N_G(v)} p\, e(u,v) + e(v,v') \;=\; s(v) + e(v,v') \;>\; s(v)$$

chips, contradicting the fact that the configuration s reappears after σ has been applied. □

Theorem 3.4 (Biggs [4]). *Let G be a connected graph, s a critical configuration on G, and $\sigma = v_1, \dots, v_k$ a legal sequence for s such that σ contains the vertex q exactly once, and such that, after applying σ to s, a stable configuration is obtained. Then $k = n$, every vertex appears exactly once in σ, and the resulting stable configuration is equal to s.*

Proof. Let σ be any legal sequence for s, which contains q exactly once (since s is stable, q must appear as the first vertex in σ), and such that a stable configuration s' is returned after applying σ. Since s is a critical configuration, there exists a legal sequence $\sigma' = v'_1, \dots, v'_\ell$ for s, and such that, after applying σ', the configuration s is returned. Note that every vertex appears the same number of times in σ', and hence every vertex appears at least once in σ'; but every vertex appears at most once in σ. This means that, for all $v \in V(G)$,

$$\max\{\, x_\sigma(v), x_{\sigma'}(v) \,\} \;=\; x_{\sigma'}(v).$$

So, if we form the sequence τ according to Lemma 2.4, then every vertex appears in τ the same number of times as the vertex does in σ'. Hence every vertex appears equally often in τ, and applying τ will result in the configuration s again. Also, we can choose τ such that its initial segment is equal to σ. Let τ' be the part of τ after the initial part σ. Then τ' is a legal sequence for s', the stable configuration resulting from applying σ on s. So now we have a sequence τ in which each vertex appears equally often: the initial part of τ is equal to σ, and each vertex appears at most once; for the remaining part τ', each vertex appears at most as often as q does, because of Corollary 3.2. This is only possible if each vertex appears exactly once in σ. But this also means that $k = n$ and $\boldsymbol{s}' = \boldsymbol{s}$, thus proving the theorem. □

Corollary 3.5. *Let $\boldsymbol{c}$ be a critical configuration. Then $0 \leqslant c(v) \leqslant d_G(v) - 1$ for all $v \in V(G) \setminus \{q\}$.*

Proof. The fact that $c(v) \leqslant d_G(v) - 1$ follows directly from the definition of a critical configuration as a special stable configuration. For the lower bound on $c(v)$, define the legal sequence σ as in Theorem 3.4. Since every vertex $u \in V(G) \setminus \{v\}$ appears once in σ, v receives

$$\sum_{u \in V(G) \setminus \{v\}} e(u, v) = d_G(v)$$

chips when applying σ. On the other hand, since v is fired as well, there must be a moment when v holds at least $d_G(v)$ chips. This is only possible if at the start v held at least 0 chips. □

Notice that Theorem 3.4 means that it is fairly easy to recognize a critical configuration: once a stable configuration $\boldsymbol{s}$ is obtained, store it in memory. Then start a sequence of legal chip firings, starting with firing q, until another stable configuration $\boldsymbol{s}'$ is obtained. Because of Lemma 3.1 this happens after all vertices are fired at most once, hence after at most n firings, so after at most $\sum_{v \in V(G)} d_G(v) = 2m$ chip movements. If $\boldsymbol{s}' = \boldsymbol{s}$, then we have found a critical configuration; otherwise forget $\boldsymbol{s}$ and repeat the procedure with $\boldsymbol{s}'$. Of course, crucial to the practical success of the approach above is some knowledge about the number of times the procedure above needs to be repeated. An upper bound for this number will be determined in the following sections.

We finish this section by showing that there exists an even more efficient way to recognize a critical configuration: we do not have to remember a candidate configuration, but only have to remember whether or not every vertex has yet been fired.

Theorem 3.6. *Let G be a connected graph and $\boldsymbol{s}$ a configuration on G. Let $\sigma = v_1, \ldots, v_k$ be a legal sequence for $\boldsymbol{s}$ chosen such that:*

(i) *applying σ to $\boldsymbol{s}$ results in a stable configuration $\boldsymbol{s}'$;*
(ii) *every vertex appears at least once in σ.*

Then $\boldsymbol{s}'$ is a critical configuration.

If, moreover, σ has been chosen such that, additionally,

(iii) *no subsequence $v_1, \ldots, v_\ell$ with $\ell < k$ has properties* (i) *and* (ii),

then, if q appears in σ, the stable configuration, that appears just before the last time q is fired when applying σ, is equal to $\boldsymbol{s}'$. This is in fact the first time a critical configuration is obtained.

Proof. We use induction on k. Suppose first that v_1 appears more than once in σ. Then we can apply the induction hypothesis on the configuration obtained after v_1 has been fired and the sequence $\sigma' = v_2, \dots, v_k$.

So we can assume that v_1 appears only once in σ. If every vertex appears only once in σ, then the configuration $\boldsymbol{s}'$ is the same as the original configuration $\boldsymbol{s}$. Since $\boldsymbol{s}'$ is a stable configuration, so is $\boldsymbol{s}$. This means that $\boldsymbol{s}$ is a critical configuration, which proves the result.

So assume that some vertex appears more than once in σ. Choose v_i such that there exists an $j > i$ with $v_j = v_i$ and such that j is minimal with that property. First suppose that $v_i = q$. Let $\overline{\boldsymbol{s}}$ be the configuration obtained from $\boldsymbol{s}$ by adding $e(v, q)$ to $s(v)$ for every $v \in N_G(q)$. (This is the configuration obtained from $\boldsymbol{s}$ by firing q if that had been legal.) Then the sequence $\overline{\sigma} = v_1, \dots, v_{i-1}, v_{i+1}, \dots, v_k$, in which each vertex appears at least once, is legal for $\overline{\boldsymbol{s}}$. Applying the induction hypothesis to $\overline{\boldsymbol{s}}$ and $\overline{\sigma}$ proves the result.

Hence we can assume $v_i \neq q$. By the choice of v_i, every vertex appears at most once in the sequence $\tau = v_1, \dots, v_{i-1}, v_i, v_{i+1}, \dots, v_{j-1}$, and when τ has been applied to $\boldsymbol{s}$, v_i is ready again. But since v_i loses $d_G(v_i)$ chips and gains at most

$$\sum_{u \in N_G(v_i)} e(u, v_i) = d_G(v_i)$$

chips, v_i must have been ready when we started applying τ. Let $\overline{\boldsymbol{s}}'$ be the configuration obtained from $\boldsymbol{s}$ by firing vertex v_i. Then the sequence $\overline{\sigma} = v_1, \dots, v_{i-1}, v_{i+1}, \dots, v_k$, in which each vertex appears at least once, is legal for $\overline{\boldsymbol{s}}'$. Now apply the induction hypothesis to $\overline{\boldsymbol{s}}'$ and $\overline{\sigma}$ to obtain the result. □

The following result, which follows directly by combining Theorems 3.4 and 3.6, is a crucial observation for later sections.

Corollary 3.7. *Let $\boldsymbol{s}$ be a configuration. Suppose that, after applying a legal sequence for $\boldsymbol{s}$ in which each vertex, except possibly q, appears at least once, a stable configuration $\boldsymbol{s}'$ is obtained. Then $\boldsymbol{s}'$ is a critical configuration.*

Corollary 3.8. *Let $\boldsymbol{c}$ be a critical configuration. Let $\boldsymbol{s}$ be a configuration with $s(v) \geqslant c(v)$ for all $v \in V(G) \setminus \{q\}$. Then the first stable configuration that appears after applying a (possibly empty) legal sequence to $\boldsymbol{s}$ is a critical configuration.*

Proof. Let $\overline{\boldsymbol{c}}$ and $\overline{\boldsymbol{s}}$ be the configurations obtained from $\boldsymbol{c}$ and $\boldsymbol{s}$, respectively, by adding $e(v, q)$ to $c(v)$ and $s(v)$ for every $v \in N_G(q)$. (These are the configurations obtained by firing q if that had been legal.) Then there exists a legal sequence $\overline{\sigma} = v_1, \dots, v_{n-1}$ for $\overline{\boldsymbol{c}}$ containing every vertex in $V(G) \setminus \{q\}$. Since $\overline{s}(v) \geqslant \overline{c}(v)$ for all $v \in V(G) \setminus \{q\}$, $\overline{\sigma}$ is also legal for $\overline{\boldsymbol{s}}$ by Lemma 2.3. Now construct a legal sequence σ for $\overline{\boldsymbol{s}}$ starting with $\overline{\sigma}$, not containing q, and resulting in a stable configuration $\overline{\boldsymbol{s}}'$. Since σ contains each vertex in $V(G) \setminus \{q\}$ at least once, $\overline{\boldsymbol{s}}'$ is a critical configuration. But after having applied $\overline{\sigma}$ on $\overline{\boldsymbol{s}}$, we have the original configuration $\boldsymbol{s}$. Applying the part of σ after $\overline{\sigma}$ to $\boldsymbol{s}$ hence leads to the first stable configuration $\overline{\boldsymbol{s}}'$, being a critical configuration. □

Corollary 3.9. *Let $\boldsymbol{c}_1, \boldsymbol{c}_2$ be two critical configurations and set $\boldsymbol{s} = \boldsymbol{c}_1 + \boldsymbol{c}_2$ (vector addition). Then the first stable configuration that appears after applying a (possibly empty) legal sequence to $\boldsymbol{s}$ is a critical configuration, and hence is equal to $\boldsymbol{c}_1 \bullet \boldsymbol{c}_2$.*

Proof. This follows directly from Corollary 3.8 upon noting that $c_2(v) \geqslant 0$ by Corollary 3.5, and hence $s(v) \geqslant c_1(v)$, for all $v \in V(G) \setminus \{q\}$. □

4. Dynamics of the dollar game

In order to obtain an answer to the question, 'How long does it take before a stable or critical configuration is reached?', we are going to look at the dollar game as some kind of dynamic process. In order to do this, we need to define a unit of time.

Let $\boldsymbol{s}$ be a configuration in the dollar game. A *cycle* for $\boldsymbol{s}$ is a sequence σ such that:

(S1) if $\boldsymbol{s}$ is not a stable configuration, then $\sigma = v_1, \dots, v_k$ is a legal sequence for $\boldsymbol{s}$ such that σ does not contain q, every other vertex appears at most once in σ, and k is as large as possible under these conditions;

(S2) if $\boldsymbol{s}$ is a stable configuration, then $\sigma = q, v_1, \dots, v_k$ is a legal sequence for $\boldsymbol{s}$ such that every vertex (including q) appears at most once in σ, and k is as large as possible under these conditions.

Note that a cycle is not uniquely determined by $\boldsymbol{s}$. But the following lemma, which is a direct consequence of Theorem 2.5, shows that some properties of a cycle are completely determined.

Lemma 4.1. *Let $\boldsymbol{s}$ be a configuration and let σ_1, σ_2 be two cycles of $\boldsymbol{s}$. Then a vertex appears the same number of times in σ_1 as in σ_2.*

In particular, it follows that the configuration $\boldsymbol{s}'$ obtained after applying a cycle to $\boldsymbol{s}$ is independent of the exact choice of the cycle, but is completely determined by $\boldsymbol{s}$.

Let $\boldsymbol{s}$ be a starting configuration in the dollar game. We call $\boldsymbol{s}$ the configuration *at time* 0 of the game, denoting $\boldsymbol{s} = \boldsymbol{s}_0$. If the configuration $\boldsymbol{s}_t$ at time t is defined, then the configuration at *time* $(t+1)$ is defined as the configuration obtained by applying one cycle to $\boldsymbol{s}_t$. Because of Lemma 4.1 this means that, for every $t \in \{0, 1, 2, 3, \dots\}$, the configuration $\boldsymbol{s}_t$ at time t is uniquely defined.

We complete this short section by giving some properties of the configurations $\boldsymbol{s}_t$.

Lemma 4.2. *Let $\boldsymbol{s}_t$ be the configuration of the dollar game at time t, defined on a graph G with n vertices and m edges.*

(a) *The configuration $\boldsymbol{s}_{t+1}$ can be obtained from $\boldsymbol{s}_t$ by firing every vertex at most once.*
(b) *Configuration $\boldsymbol{s}_{t+1}$ can be obtained from $\boldsymbol{s}_t$ in at most n firings.*
(c) *Configuration $\boldsymbol{s}_{t+1}$ can be obtained from $\boldsymbol{s}_t$ in at most $2m$ chip movements.*

Proof. Statements (a) and (b) follow directly from the definition of a cycle for $\boldsymbol{s}_t$ in (S1) or (S2). For (c), observe that firing a vertex $v \in V(G)$ involves $d_G(v)$ chip movements. Hence firing every vertex at most once involves at most $\sum_{v \in V(G)} d_G(v) = 2m$ chip movements. □

Lemma 4.3. *Let $\boldsymbol{s}_0$ be the starting configuration of a dollar game on a graph G. Then, for any $v \in V(G) \setminus \{q\}$ and $t \in \{0, 1, 2, \dots\}$, we have:*

(a) *if $s_t(v) < 0$, then $s_t(v) \leqslant s_{t'}(v) \leqslant d_G(v) - 1$ for all $t' \geqslant t$;*
(b) *if $0 \leqslant s_t(v) \leqslant d_G(v) - 1$, then $0 \leqslant s_{t'}(v) \leqslant d_G(v) - 1$ for all $t' \geqslant t$;*
(c) *if $s_t(v) \geqslant d_G(v)$, then $0 \leqslant s_{t'}(v) \leqslant s_t(v)$ for all $t' \geqslant t$.*

Proof. Let $v \in V(G) \setminus \{q\}$ and $t \in \{0, 1, 2, \dots\}$, and let σ be a cycle for $\boldsymbol{s}_t$. Since every vertex $u \in V(G) \setminus \{v\}$ appears at most once in σ, v receives at most

$$\sum_{u \in V(G) \setminus \{v\}} e(u, v) = d_G(v)$$

chips when applying the cycle to $\boldsymbol{s}_t$. On the other hand, v itself is only fired if $s(v) \geqslant d_G(v)$ for a certain configuration, and then v loses $d_G(v)$ chips during the cycle.

If $s_t(v) < 0$, then v is not fired in the cycle σ, and hence $s_t(v) \leqslant s_{t+1}(v) \leqslant s_t(v) + d_G(v) < d_G(v)$. If $s_t(v) \geqslant d_G(v)$, then v is certainly fired in the cycle, which gives $0 \leqslant s_t(v) - d_G(v) \leqslant s_{t+1}(v) \leqslant s_t(v)$. And if $0 \leqslant s_t(v) \leqslant d_G(v) - 1$, and v is not fired in the cycle σ, then $0 \leqslant s_t(v) \leqslant s_{t+1}(v) < d_G(v)$. Finally, if $0 \leqslant s_t(v) \leqslant d_G(v) - 1$, and while applying the cycle σ we obtain a configuration $\boldsymbol{s}$ with $s(v) \geqslant d_G(v)$, then v will be fired in the cycle and hence $0 \leqslant s_{t+1}(v) \leqslant s_t(v) + d_G(v) - d_G(v) = s_t(v) \leqslant d_G(v) - 1$.

The result follows by setting $t' = t + i$ and applying induction on i. □

5. Basic theory of the oil game

Our main tool in analysing the discrete dollar game will be a continuous version of the game. In order to develop intuition for what is happening in this game, one can consider it as an *oil game*, in which quantities of oil instead of chips are transported from one vertex of the graph to another.

As in the dollar game we will assume that we are given a finite, undirected, connected graph G and one special vertex $q \in V(G)$. At a certain time we assume that each vertex $v \in V(G) \setminus \{q\}$ contains a quantity $r(v)$ of oil, which can be negative. An $(n-1)$-vertex $\boldsymbol{r}$ of all oil quantities is called a *configuration* of the oil game. The flow of oil out of a vertex v is indicated by $\varphi(v)$, where $\varphi(v)$ is always nonnegative. We interpret $\varphi(v)$ as the amount of oil per unit of time that is pumped away from v through each of the edges incident with v. The value of $\varphi(v)$, given a configuration $\boldsymbol{r}$, will be determined by the following set of rules.

(O1a) If $v \in V(G) \setminus \{q\}$ with $r(v) < 0$, then no oil will be pumped away from v; hence $\varphi(v) = 0$.
(O1b) If $v \in V(G) \setminus \{q\}$ with $r(v) > 0$, then a flow of one unit of oil per unit of time per edge will be pumped away from v; hence $\varphi(v) = 1$.
(O1c) If $v \in V(G) \setminus \{q\}$ with $r(v) = 0$, then all input of oil that v receives from its neighbours will be pumped away from v as well, evenly distributed over all edges incident with v.
(O2a) If $v = q$ and there exists a vertex $u \in V(G) \setminus \{q\}$ with $r(u) > 0$, then q will output no oil; hence $\varphi(q) = 0$.

(O2b) If $v = q$ and $r(u) \leqslant 0$ for all $u \in V(G) \setminus \{q\}$, then one unit of oil per unit of time per edge will be pumped away from q; hence $\varphi(q) = 1$.

Another way to describe the oil game is by saying that the oil flow in a game with configuration $\boldsymbol{r}$ is determined by the solutions $\varphi(v)$, $v \in V(G)$, of the following collection of equations:

$$\varphi(v) = \begin{cases} 0, & \text{if } r(v) < 0, \\ \dfrac{1}{d_G(v)} \displaystyle\sum_{u \in N_G(v)} e(u,v)\,\varphi(u), & \text{if } r(v) = 0, \\ 1, & \text{if } r(v) > 0, \end{cases} \qquad \text{for all } v \in V(G) \setminus \{q\}; \tag{5.1}$$

$$\varphi(q) = \begin{cases} 0, & \text{if there exists a } u \in V(G) \setminus \{q\} \text{ with } r(u) > 0, \\ 1, & \text{if for all } u \in V(G) \setminus \{q\} \text{ we have } r(u) \leqslant 0. \end{cases} \tag{5.2}$$

We say that a vertex $v \in V(G) \setminus \{q\}$ is *passive* if $r(v) < 0$, *saturated* if $r(v) = 0$, and *active* if $r(v) > 0$. The vertex q is *active* if no other vertex is active, and *passive* otherwise. For a certain configuration $\boldsymbol{r}$, let $V_{\boldsymbol{r}}^a$ be the set of active vertices and $V_{\boldsymbol{r}}^p$ the set of passive vertices.

Looking at the equations for φ in (5.1) and (5.2), it appears that we can interpret φ as an electrical flow in a certain electrical network, as described below. See, *e.g.*, [10, Chapters II and IX] for definitions and the theory of electrical networks needed.

Proposition 5.1. *The oil flow $\varphi(v)$, $v \in V(G)$, in a connected graph G with configuration $\boldsymbol{r}$ is equal to the potential solution of the electrical network on the vertices of G in which the conductance between two vertices u, v is $e(u,v)$, the vertices in $V_{\boldsymbol{r}}^p$ have potential 0, the vertices in $V_{\boldsymbol{r}}^a$ have potential 1, and the vertices in $V(G) \setminus (V_{\boldsymbol{r}}^a \cup V_{\boldsymbol{r}}^p)$ satisfy Kirchhoff's currency law, that is,*

$$d_G(v)\,\varphi(v) = \sum_{u \in N_G(v)} e(u,v)\,\varphi(u), \qquad \textit{for} \;\; v \in V(G) \setminus (V_{\boldsymbol{r}}^a \cup V_{\boldsymbol{r}}^p).$$

Note in particular that the theory of electrical networks guarantees a unique solution to the potential problem on a connected graph. So we get that the oil flow φ is uniquely determined by equations (5.1) and (5.2).

Lemma 5.2. *For a configuration $\boldsymbol{r}$, we have that $0 \leqslant \varphi(v) \leqslant 1$ for all $v \in V(G)$.*

Proof. Suppose that there exists a configuration $\boldsymbol{r}$ for which $\varphi(v) > 1$ for some $v \in V(G)$. Choose v such that $\varphi(v)$ is maximum. Because of equation (5.2), this means that $v \neq q$. According to (5.1) we find that $r(v) = 0$ and

$$\varphi(v) = \frac{1}{d_G(v)} \sum_{u \in N_G(v)} e(u,v)\,\varphi(u).$$

Since $\varphi(u) \leqslant \varphi(v)$ for all $u \in V(G)$, this gives

$$d_G(v)\,\varphi(v) \;=\; \sum_{u \in N_G(v)} e(u,v)\,\varphi(u) \;\leqslant\; \sum_{u \in N_G(v)} e(u,v)\,\varphi(v) \;=\; d_G(v)\,\varphi(v).$$

So we must have equality; in particular it must hold that $\varphi(u) = \varphi(v)$ for all $u \in N_G(v)$. Continuing the same reasoning for the neighbours of v, and using that G is connected, we obtain that $\varphi(u) = \varphi(v) > 1$ for all $u \in V(G)$. This contradicts the fact that $\varphi(q) \in \{0, 1\}$.

A similar argument gives a contradiction if $\varphi(v) < 0$ for some $v \in V(G)$. □

6. Dynamics of the oil game

The picture of the oil game sketched in the previous section is that of a static game. We will now add a dynamical element, interpreting the oil flows φ as quantities of oil that are added or removed from the amounts of oil $r(v)$ at a vertex v. In order to emphasize this dynamical behaviour we index the variables with the time t: $\boldsymbol{r}_t$ and φ_t. The values of φ_t, given $\boldsymbol{r}_t$, are determined by equations (5.1) and (5.2). The dynamics of $\boldsymbol{r}_t$ will be determined by the equation

$$\frac{d}{dt} r_t(v) \;=\; -d_G(v)\,\varphi_t(v) + \sum_{u \in N_G(v)} e(u,v)\,\varphi_t(u), \qquad \text{for all } v \in V(G) \setminus \{q\}. \tag{6.1}$$

We usually assume that the game starts at time $t = 0$ with a configuration $\boldsymbol{r}_0$.

Lemma 6.1. *For any $v \in V(G) \setminus \{q\}$ and $t \geqslant 0$ we have:*

(a) *if $r_t(v) < 0$, then $r_t(v) \leqslant r_{t'}(v) \leqslant 0$ for all $t' \geqslant t$;*
(b) *if $r_t(v) = 0$, then $r_t(v) = r_{t'}(v) = 0$ for all $t' \geqslant t$;*
(c) *if $r_t(v) > 0$, then $0 \leqslant r_{t'}(v) \leqslant r_t(v)$ for all $t' \geqslant t$.*

Proof. If $r_t(v) = 0$, then equations (5.1) and (6.1) give that $\frac{d}{dt} r_t(v) = 0$; hence $r_{t'}(v) = r_t(v) = 0$ for all $t' \geqslant t$.

If $r_t(v) < 0$, then $\varphi_t(v) = 0$ by equation (5.1). From Lemma 5.2 we learn that $\varphi_t(u) \geqslant 0$ for all $u \in V(G)$ Using equation (6.1) this means that

$$\frac{d}{dt} r_t(v) = -d_G(v) \cdot 0 + \sum_{u \in N_G(v)} e(u,v)\,\varphi_t(u) \geqslant 0,$$

and hence $r_{t'}(v) \geqslant r_t(v)$ for $t' \geqslant t$ as long as $r_{t'}(v) < 0$. But once $r_{t'}(v) = 0$, we get that $r_{t''}(v) = r_{t'}(v) = 0$ for $t'' \geqslant t'$.

If $r_t(v) > 0$, then we can do a similar reasoning using that $\varphi_t(v) = 1$ and $\varphi_t(u) \leqslant 1$ for all $u \in V(G)$, hence

$$\frac{d}{dt} r_t(v) = -d_G(v) \cdot 1 + \sum_{u \in N_G(v)} e(u,v)\,\varphi_t(u) \leqslant 0. \qquad \square$$

An *active configuration* of the oil game is a configuration $\boldsymbol{r}$ in which $r(v) > 0$ for some $v \in V(G) \setminus \{q\}$; otherwise the configuration is called *inactive*. Note that the vertex q is passive in an active configuration and active in an inactive configuration. A *recurrent configuration* of the oil game is a configuration such that the total inflow and outflow is the same for each vertex. In particular, if $\boldsymbol{r}_t$ is a recurrent configuration, then $\frac{d}{dt}\boldsymbol{r}_t = \boldsymbol{0}$ by equation (6.1) and hence $\boldsymbol{r}_{t'} = \boldsymbol{r}_t$ for all $t' \geqslant t$. So a recurrent configuration of the oil game can be considered as some kind of 'critical configuration' of the oil game.

The following are the two crucial results concerning recurrent configurations of the oil game. Recall that $\|\boldsymbol{r}\| = \sum_{v\in V(G)\setminus\{q\}}|r(v)|$. Define

$$\|\boldsymbol{r}\|^{+} = \sum_{v\in V(G)\setminus\{q\},\, r(v)>0} r(v) = \sum_{v\in V_{\boldsymbol{r}}^{a}\setminus\{q\}} r(v),$$
$$\|\boldsymbol{r}\|^{-} = \sum_{v\in V(G)\setminus\{q\},\, r(v)<0} |r(v)| = \sum_{v\in V_{\boldsymbol{r}}^{p}\setminus\{q\}} -r(v).$$

Theorem 6.2. *For a connected graph G, the only recurrent configuration of the oil game is the configuration $\boldsymbol{r}^0 = \mathbf{0}$. For the oil game in the recurrent configuration $\boldsymbol{r}^0$, we have that $\varphi(v) = \mathbf{1}$.*

Theorem 6.3. *Let G be a connected graph on n vertices and with edge-connectivity λ, and let $\boldsymbol{r}_0$ be a starting configuration of the oil game on G.*

(a) *If $\boldsymbol{r}_0$ is active, then, for any $t \geqslant 3\,n\,\|\boldsymbol{r}_0\|^{+}/(\lambda+1)$, $\boldsymbol{r}_t$ is a passive configuration.*
(b) *If $\boldsymbol{r}_0$ is passive, then, for any $t \geqslant 3\,n\,\|\boldsymbol{r}_0\|^{-}/(\lambda+1)$, $\boldsymbol{r}_t = \boldsymbol{r}^0$.*
(c) *For any $t \geqslant 3\,n\,\|\boldsymbol{r}_0\|/(\lambda+1)$ we have that $\boldsymbol{r}_t = \boldsymbol{r}^0$.*

Theorem 6.3 will follow from results later in this section.

Proof of Theorem 6.2. It is obvious that $\boldsymbol{r}^0$ is a recurrent configuration with $\varphi(v) = 1$ for all $v \in V(G)$.

Let $\boldsymbol{r}_t$ be a configuration in which some vertices in $V(G)\setminus\{q\}$ are active; hence $\varphi_t(v) = 1$ for some $v \in V(G)\setminus\{q\}$. Since G is connected and $\varphi_t(q) = 0$, there must be vertices $v \in V(G)\setminus\{q\}$ and $u \in N_G(v)$, such that $\varphi_t(v) = 1$ and $\varphi_t(u) < 1$. Then $\frac{d}{dt}r_t(v) < 0$, and hence $\boldsymbol{r}$ cannot be a recurrent configuration.

If $\boldsymbol{r}_t$ is a configuration in which q is active, hence $\varphi_t(q) = 1$, but not all vertices are saturated, then we can find vertices $v \in V(G)\setminus\{q\}$ and $u \in N_G(v)$, such that $\varphi_t(v) = 0$ and $\varphi_t(u) > 0$. Then $\frac{d}{dt}r_t(v) > 0$, and again we must conclude that $\boldsymbol{r}$ is not a recurrent configuration. □

Another way to phrase Theorem 6.2 is to say that a configuration is not recurrent as long as there are passive vertices. It also follows from the proof of the theorem that, in a non-recurrent configuration, there is a net oil flow from the active to the passive vertices. It is the amount of this net flow which will be the crucial parameter that determines how long it takes before we reach the recurrent configuration $\boldsymbol{r}^0$.

Given a configuration $\boldsymbol{r} \neq \boldsymbol{r}^0$ on G, the graph G^* is obtained from G by contracting all vertices of $V_{\boldsymbol{r}}^{a}$ into one vertex a^* and similarly all vertices of $V_{\boldsymbol{r}}^{p}$ into one vertex p^*, removing loops but not multiple edges. If $\boldsymbol{r}$ is inactive (hence $q \in V_{\boldsymbol{r}}^{a}$), then we set $q^* = a^*$ and $r^*(p^*) = \sum_{v\in V_{\boldsymbol{r}}^{p}} r(v)$; similarly, if $\boldsymbol{r}$ is active (hence $q \in V_{\boldsymbol{r}}^{p}$), then we set $q^* = p^*$ and $r^*(a^*) = \sum_{v\in V_{\boldsymbol{r}}^{a}} r(v)$. For vertices $v \in V(G)\setminus(V_{\boldsymbol{r}}^{a}\cup V_{\boldsymbol{r}}^{p})$ we set $r^*(v) = r(v)$.

The following lemma can be proved using the theory of electrical networks, using Proposition 5.1. For completeness we give a more intuitive proof.

Lemma 6.4. *For a non-recurrent configuration $\boldsymbol{r}$, the net flow in G from the active to the passive vertices is the same as the net flow in G^*, with configuration $\boldsymbol{r}^*$, from the unique active vertex in G^* to the unique passive vertex in G^*.*

Proof. If there exists an edge in G between two passive vertices, then no flow occurs along this edge. Similarly, for an edge between two active vertices, we have that a flow of size 1 goes in both directions; hence no net flow goes through such an edge. So we can remove any of this type of edge without changing the flow pattern in the remainder of the graph. Thus we can assume that there are no edges in G between vertices in $V_{\boldsymbol{r}}^a$ and between vertices in $V_{\boldsymbol{r}}^p$. Next suppose that $\boldsymbol{r}$ is active. If we identify two vertices from $V_{\boldsymbol{r}}^a$ forming one new vertex, adapting $\boldsymbol{r}$ in the appropriate way, then again we see that this does not change the flow pattern, since the flow out of the new vertex is the same as the total flow out of the two original vertices. A similar observation applies when different vertices from $V_{\boldsymbol{r}}^p$ are identified. Continuing with this identification process, we eventually obtain the graph G^* with configuration $\boldsymbol{r}^*$ in which the same flow pattern appears as it did in G with configuration $\boldsymbol{r}$. In particular, the total net flow from the active to the passive vertices is the same before and after the contraction. □

Given a graph G with non-recurrent configuration $\boldsymbol{r}$, let φ^* be the flow in the graph G^* with configuration $\boldsymbol{r}^*$. The following is a translation of Lemma 6.4, using equation (6.1).

Corollary 6.5. *For a non-recurrent configuration $\boldsymbol{r}_t$ on G at time t, we have the following properties.*

(a) *It holds that $\varphi_t^*(a^*) = 1$. Moreover, if $\boldsymbol{r}_t$ is active (hence $q \notin V_{\boldsymbol{r}}^a$ and $a^* \neq q^*$), then*

$$\begin{aligned}\frac{d}{dt} r_t^*(a^*) &= -d_{G^*}(a^*) + \sum_{u \in N_{G^*}(a^*)} e(u,a^*)\, \varphi_t^*(u) \\ &= \sum_{v \in V_{\boldsymbol{r}_t}^a} \Big[-d_G(v) + \sum_{u \in N_G(v)} e(u,v)\, \varphi(v) \Big] = \sum_{v \in V_{\boldsymbol{r}_t}^a} \frac{d}{dt} r_t(v).\end{aligned}$$

(b) *It holds that $\varphi_t^*(p^*) = 0$. Moreover, if $\boldsymbol{r}_t$ is inactive (hence $q \notin V_{\boldsymbol{r}}^p$ and $p^* \neq q^*$), then*

$$\frac{d}{dt} r_t^*(p^*) = \sum_{u \in N_{G^*}(p^*)} e(u,p^*)\, \varphi_t^*(u) = \sum_{v \in V_{\boldsymbol{r}_t}^p} \Big[\sum_{u \in N_G(v)} e(u,v)\, \varphi(v) \Big] = \sum_{v \in V_{\boldsymbol{r}_t}^p} \frac{d}{dt} r_t(v).$$

The main advantage of Lemma 6.4 and Corollary 6.5 is that an instance of the oil problem, with possibly many active and/or passive vertices, is translated into an instance with only one active and one passive vertex. This will prove a major advantage once we return to the relationship between oil flows and electrical networks.

For the graph G^* define $R_{\text{eff}}(a^*, p^*)$ as the *effective resistance* between a^* and p^*, *i.e.*, the resistance between a^* and p^* in the electrical network given by the graph G^* if all edges are assumed to be resistors with resistance one. Here we assume that multiple edges appear as multiple resistors. Another way to obtain the same is by replacing an edge with multiplicity e by one resistor with resistance $1/e$. Define the *effective conductance* $C_{\text{eff}}(a^*, p^*)$ as $1/R_{\text{eff}}(a^*, p^*)$.

Theorem 6.6. *Let $\boldsymbol{r}_t$ be a non-recurrent configuration on a connected graph G at time t.*

(a) *If $\boldsymbol{r}_t$ is active, then* $\sum_{v \in V^a_{\boldsymbol{r}_t}} \frac{d}{dt} r_t(v) = -C_{\text{eff}}(a^*, p^*)$.

(b) *If $\boldsymbol{r}_t$ is inactive, then* $\sum_{v \in V^p_{\boldsymbol{r}_t}} \frac{d}{dt} r_t(v) = C_{\text{eff}}(a^*, p^*)$.

Proof. We only prove the case that $\boldsymbol{r}_t$ is active, the other case being similar. The flow φ^*_t is the solution of the potential problem on the network represented by the graph G^* in which every edge has unit resistance, where we set $\varphi^*_t(a^*) = 1$, $\varphi^*_t(p^*) = 0$, and where we require that Kirchhoff's currency law holds for all other vertices. This means that the electrical flow from a^* to a neighbouring vertex $u \in N_{G^*}(a^*)$ through the $e(u, a^*)$ edges connecting u and a^* is equal to $e(u, a^*) \cdot (\varphi^*_t(a^*) - \varphi^*_t(u))$. (Recall that every edge has resistance one.) So the total electrical flow away from a^* is equal to

$$\begin{aligned} \sum_{u \in N_{G^*}(a^*)} e(u, a^*) \cdot (\varphi^*_t(a^*) - \varphi^*_t(u)) &= \sum_{u \in N_{G^*}(a^*)} e(u, a^*) \cdot 1 - \sum_{u \in N_{G^*}(a^*)} e(u, a^*)\, \varphi^*_t(u) \\ &= d_{G^*}(a^*) - \sum_{u \in N_{G^*}(a^*)} e(u, a^*)\, \varphi^*_t(u). \end{aligned}$$

This flow is equal to the flow from a^* to p^*, hence equal to $C_{\text{eff}}(a^*, p^*)$, and the result follows from Corollary 6.5(a). □

In order to use Theorem 6.6 we want to have a lower bound for $C_{\text{eff}}(a^*, p^*)$. Several of these lower bounds exist in the literature. However, many of them involve the degrees of the graph under consideration. Since we are working with the graph G^*, which can have degrees that are quite different from G, we need to do a little translation to get a lower bound depending on G only. The following is implicit in the proof of [12, Theorem 6].

Lemma 6.7 (Coppersmith, Feige and Shearer [12]). *For any pair of vertices s and t in a connected graph H, the effective conductance between s and t satisfies*

$$C_{\text{eff}}(s, t) > \left[3 \sum_{v \in V(H)} \frac{1}{d_H(v) + 1} \right]^{-1}.$$

Corollary 6.8. *Let $\boldsymbol{r}_t$ be a non-recurrent configuration on the graph G with n vertices and edge-connectivity λ, and let G^* be as defined above Lemma 6.4. Then the effective conductance between a^* and p^* in G^* satisfies $C_{\text{eff}}(a^*, p^*) > \frac{\lambda+1}{3n}$.*

Proof. Let G^* have n^* vertices and edge-connectivity λ^*. Since $d_{G^*}(v) \geqslant \lambda^*$ for all $v \in V(G^*)$, we find from Lemma 6.7 that $C_{\text{eff}}(a^*, p^*) > \frac{\lambda^*+1}{3n^*}$. Since contraction does not reduce the edge-connectivity, we have $\lambda^* \geqslant \lambda$. Since trivially $n^* \leqslant n$, the result follows. □

We now can give the proof of Theorem 6.3.

Proof of Theorem 6.3. Let $\boldsymbol{r}_t$ be an active configuration on G. From Theorem 6.6 and Corollary 6.8 it follows that

$$\sum_{v \in V^a_{\boldsymbol{r}_t}} \frac{d}{dt} r_t(v) < -\frac{\lambda+1}{3\,n}.$$

Note that this bound is independent of the set of vertices $V^a_{\boldsymbol{r}}$, although this set will change over time. In particular we find that

$$\sum_{v \in V^a_{\boldsymbol{r}_t}} r_t(v) \;<\; \sum_{v \in V^a_{\boldsymbol{r}_0}} r_0(v) - t \cdot \frac{\lambda+1}{3\,n} \;=\; \|\boldsymbol{r}_0\|^+ - t \cdot \frac{\lambda+1}{3\,n}.$$

Since for an active configuration $\boldsymbol{r}$ we must have $\sum_{v \in V^a_{\boldsymbol{r}}} r(v) > 0$, it is impossible that the configuration is still active for $t \geqslant 3\,n\,\|\boldsymbol{r}_0\|^+/(\lambda+1)$, thus proving part (a).

The proof of (b) is similar.

So let $\boldsymbol{r}_0$ be any configuration. Let t^p be the moment in time in which the configuration turns passive, where it is possible that $t^p = 0$. From (b) we see that, for any $t \geqslant t^p + 3\,n\,\|\boldsymbol{r}_{t^p}\|^-/(\lambda+1)$, $\boldsymbol{r}_t$ is the recurrent configuration. Hence, if $t \geqslant 3\,n\,\|\boldsymbol{r}_0\|^+/(\lambda+1) + 3\,n\,\|\boldsymbol{r}_{t^p}\|^-/(\lambda+1)$, then $\boldsymbol{r}_t = \boldsymbol{r}^0$. From Lemma 6.1 we obtain $\|\boldsymbol{r}_{t^p}\|^- \leqslant \|\boldsymbol{r}_0\|^-$, which gives $\|\boldsymbol{r}_0\| = \|\boldsymbol{r}_0\|^+ + \|\boldsymbol{r}_0\|^- \geqslant \|\boldsymbol{r}_0\|^+ + \|\boldsymbol{r}_{t^p}\|^-$. So, if $t \geqslant 3\,n\,\|\boldsymbol{r}_0\|/(\lambda+1)$, then $\boldsymbol{r}_t = \boldsymbol{r}^0$, which completes the proof of Theorem 6.3. □

7. From a configuration in the dollar game to a critical configuration

Theorem 6.3 gives upper bounds on the time it takes before an initial configuration in the oil game reaches the recurrent state. In this section we obtain similar results for the dollar game. The main tools will be results that show connections between instances of the dollar game on a graph and certain instances of the oil game on the same graph. Applying Theorem 6.3 then makes it possible to give upper bounds on the time it takes before an initial configuration in the dollar game reaches its critical state. (Here 'time' is used in the sense of Section 4.) Once we have these results, we can finally obtain the results announced in the Introduction.

We begin with some additional notation. For a starting configuration $\boldsymbol{r}_0$ of the oil game on a graph G, a vertex $v \in V(G)$, and any *real number* $t \geqslant 0$, define $\varphi_t(v)$ and $\boldsymbol{r}_t$ as in Section 6. Also define $\Phi_t(v) = \int_0^t \varphi_x(v)\,dx$. Because of Lemma 5.2, $0 \leqslant \Phi_t(v) \leqslant t$.

For a starting configuration $\boldsymbol{s}_0$ of the dollar game on G, a vertex $v \in V(G)$, and any *integer* $t \geqslant 0$, define $\boldsymbol{s}_t$ as in Section 4. For $t \geqslant 1$ define $\psi_t(v)$ as the number of times that vertex v has been fired when going from configuration $\boldsymbol{s}_{t-1}$ to $\boldsymbol{s}_t$ (hence $\psi_t(v) \in \{0, 1\}$), and define $\Psi_t(v) = \sum_{z=1}^t \psi_z(v)$. Set $\psi_0(v) = \Psi_0(v) = 0$. We again obtain $0 \leqslant \Psi_t(v) \leqslant t$ for all $t \geqslant 0$.

We can interpret $\Phi_t(v)$ as the total amount of oil that has flowed away from vertex v through any edge incident with v during the time from 0 to t. Similarly, $\Psi_t(v)$ is the total number of chips that have moved away from v along each edge incident with v between time 0 and time t. From the rules for the dollar game and the oil game, for all integers $t \geqslant 0$, we get

$$\begin{aligned} s_t(v) &= s_0(v) - d_G(v)\,\Psi_t(v) + \sum_{u\in N_G(v)} e(u,v)\,\Psi_t(u), \\ r_t(v) &= r_0(v) - d_G(v)\,\Phi_t(v) + \sum_{u\in N_G(v)} e(u,v)\,\Phi_t(u), \end{aligned}$$

which gives

$$\begin{aligned} &s_t(v) - r_t(v) \\ &\quad = s_0(v) - r_0(v) - d_G(v)\,(\Psi_t(v) - \Phi_t(v)) + \sum_{u\in N_G(v)} e(u,v)\,(\Psi_t(u) - \Phi_t(u)). \end{aligned} \tag{7.1}$$

The next two lemmas form the key results connecting the dollar game and the oil game.

Lemma 7.1. *Let $\boldsymbol{s}_0$ be a starting configuration for the dollar game on the graph G, and suppose that $\boldsymbol{s}_0$ is stable. Define the starting configuration $\boldsymbol{r}_0$ for the oil game on G by $r_0(v) = s_0(v) - d_G(v) + 1$ for $v \in V(G) \setminus \{q\}$. Then, for all $v \in V(G)$ and integers $t \geqslant 0$, we have $\Phi_t(v) \leqslant \Psi_t(v)$.*

Proof. Suppose the result is false, so that $\Phi_t(v) > \Psi_t(v)$ for some $v \in V(G)$ and $t \geqslant 0$. Choose $v \in V(G)$ such that $\Psi_t(v) - \Phi_t(v) < 0$ is minimal. It follows from Lemma 4.3(a), (b), that for every integer $z \geqslant 0$ the configuration $\boldsymbol{s}_z$ is stable, and hence $\psi_z(q) = 1$. This gives $\Psi_t(q) = t$. By Lemma 5.2 we have that $\Phi_t(q) \leqslant t$. This means that v cannot be equal to q.

Since $\boldsymbol{s}_0$ is a stable configuration, we have $s_0(v) \leqslant d_G(v) - 1$; hence $r_0(v) \leqslant 0$ and in fact $r_x(v) \leqslant 0$ for all $x \geqslant 0$ by Lemma 6.1(a), (b). If $r_t(v) < 0$, then by Lemma 6.1(a) also $r_x(v) < 0$ for all real numbers x with $0 \leqslant x \leqslant t$. But then $\varphi_x(v) = 0$ for all $0 \leqslant x \leqslant t$ and hence $\Phi_t(v) = 0$, contradicting that $\Phi_t(v) > \Psi_t(v) \geqslant 0$. We conclude that $r_t(v) = 0$.

Since v was chosen such that $\Psi_t(u) - \Phi_t(u) \geqslant \Psi_t(v) - \Phi_t(v)$ for all $u \in V(G)$, we get from equation (7.1) that

$$\begin{aligned} s_t(v) - r_t(v) &\geqslant s_0(v) - r_0(v) + \Bigg(\sum_{u\in N_G(v)} e(u,v) - d_G(v)\Bigg)(\Psi_t(v) - \Phi_t(v)) \\ &= s_0(v) - r_0(v). \end{aligned}$$

Because $r_t(v) = 0$ and $s_0(v) - r_0(v) = d_G(v) - 1$, this means $s_t(v) \geqslant d_G(v) - 1$. On the other hand, by Lemma 4.3(a), (b) we have that $s_t(v) \leqslant d_G(v) - 1$. Hence we must have equality in every inequality used so far. In particular we find that $\Psi_t(u) - \Phi_t(u) = \Psi_t(v) - \Phi_t(v) < 0$ for all $u \in N_G(v)$. Using that G is connected, we can continue the reasoning to conclude that $\Psi_t(u) - \Phi_t(u) < 0$ for all $u \in V(G)$. But this contradicts the observation in the first paragraph of this proof that $\Psi_t(q) - \Phi_t(q) \geqslant 0$. □

Lemma 7.2. *Let $\boldsymbol{s}_0$ be a starting configuration for the dollar game on the graph G, and suppose that $\boldsymbol{s}_0$ is not stable. Define the starting configuration $\boldsymbol{r}_0$ for the oil game on G by $r_0(v) = s_0(v)$ for $v \in V(G) \setminus \{q\}$.*

(a) *For all integers $t \geqslant 0$ such that s_t is not stable, we have that $\Psi_t(v) \leqslant \Phi_t(v)$ for all $v \in V(G)$.*
(b) *For all integers $t \geqslant 0$ such that s_t is not stable, if $s_t(v) \geqslant d_G(v)$ for a $v \in V(G) \setminus \{q\}$, then $s_t(v) \leqslant r_t(v)$.*

Proof. Suppose that (a) is false, so that $\Psi_t(v) > \Phi_t(v)$ for some $t \geqslant 0$ and $v \in V(G)$ while $\boldsymbol{s}_t$ is still not stable. Choose $v \in V(G)$ such that $\Psi_t(v) - \Phi_t(v) > 0$ is maximal. It follows from Lemma 4.3(c) that $\boldsymbol{s}_z$ is not stable, and so $\psi_z(q) = 0$, for every $z \in \{0, 1, \ldots, t\}$. This gives that $\Psi_t(q) = 0 \leqslant \Phi_t(q)$, and hence $v \neq q$.

Since $\Psi_t(v) > \Phi_t(v) \geqslant 0$, we must have that $\psi_z(v) = 1$ for some $z \in \{1, 2, \ldots, t\}$ and hence $s_{z-1}(v) \geqslant 0$. By Lemma 4.3(b), (c) this means that $s_t(v) \geqslant 0$ as well. Also, because of $\Phi_t(v) < \Psi_t(v) \leqslant t$, we must have that $\varphi_x(v) = 0$ for some real number x, with $0 \leqslant x \leqslant t$ and hence $r_x(v) \leqslant 0$. By Lemma 6.1(a), (b) this means $r_t(v) \leqslant 0$ as well.

Now recall that v was chosen such that $\Psi_t(u) - \Phi_t(u) \leqslant \Psi_t(v) - \Phi_t(v)$ for all $u \in V(G)$. Using this in equation (7.1) gives $s_t(v) - r_t(v) \leqslant s_0(v) - r_0(v) = 0$, and so $s_t(v) \leqslant r_t(v) \leqslant 0$. But since $s_t(v) \geqslant 0$, we must have equality in every inequality used so far. In particular we find that $\Psi_t(u) - \Phi_t(u) = \Psi_t(v) - \Phi_t(v) > 0$ for all $u \in N_G(v)$. Using that G is connected, we can continue the reasoning to conclude that $\Psi_t(u) - \Phi_t(u) > 0$ for all $u \in V(G)$. But this contradicts the observation in the first paragraph of this proof that $\Psi_t(q) - \Phi_t(q) \leqslant 0$. This completes the proof of part (a).

For part (b), let $t \geqslant 0$ be an integer such that $\boldsymbol{s}_t$ is not stable and suppose $v \in V(G) \setminus \{q\}$ with $s_t(v) \geqslant d_G(v)$. Because of Lemma 4.3(a), (b) this gives that $s_z(v) \geqslant d_G(v)$ for all $z \in \{0, 1, \ldots, t\}$. Hence $\psi_z(v) = 1$ for all $z \in \{1, 2, \ldots, t\}$ and so $\Psi_t(v) = t \geqslant \Phi_t(v)$. Using part (a), this means $\Psi_t(v) = \Phi_t(v)$. Combining this with the knowledge from (a) that $\Psi_t(u) - \Phi_t(u) \leqslant 0$ for all $u \in V(G)$, we get from equation (7.1) that $s_t(v) - r_t(v) \leqslant s_0(v) - r_0(v) = 0$, which proves part (b). □

We are now ready to prove the most important theorem in this paper.

Theorem 7.3. *Let $\boldsymbol{s}_0$ be a starting configuration of the dollar game on the graph G with n vertices, m edges, and edge-connectivity λ.*

(a) *Then, for any integer $t \geqslant 3\,n\,\|\boldsymbol{s}_0\|^+/(\lambda + 1)$, $\boldsymbol{s}_t$ is a stable configuration.*
(b) *And, for any integer $t \geqslant 3\,n\,(\|\boldsymbol{s}_0\| + 2\,m)/(\lambda + 1) + 1$, $\boldsymbol{s}_t$ is a critical configuration.*

Proof. Define the starting configuration $\boldsymbol{r}_0$ of the oil game on G as in Lemma 7.2 and let $t \geqslant 3\,n\,\|\boldsymbol{s}_0\|^+/(\lambda + 1)$ be an integer. Since $\|\boldsymbol{s}_0\|^+ = \|\boldsymbol{r}_0\|^+$, Theorem 6.3(a) gives that $\boldsymbol{r}_t$ is a passive configuration of the oil game. Hence $r_t(v) \leqslant 0$ for all $v \in V(G) \setminus \{q\}$. Following Lemma 7.2(b), this means that $\boldsymbol{s}_t$ is stable, or $s_t(v) \leqslant d_G(v) - 1$ or $s_t(v) \leqslant r_t(v) \leqslant 0$ for all $v \in V(G) \setminus \{q\}$. All possibilities lead to the conclusion that $\boldsymbol{s}_t$ is stable, thus proving part (a).

Let $t^s \geqslant 0$ be the smallest integer such that $\boldsymbol{s}_{t^s}$ is stable. Define the configuration $\boldsymbol{r}_0$ for the oil game by setting $r_0(v) = s_{t^s}(v) - d_G(v) + 1$ for all $v \in V(G) \setminus \{q\}$. From Theorem 6.3(b) it follows that if $t' \geqslant 3\,n\,\|\boldsymbol{r}_0\|^-/(\lambda + 1)$, then $\boldsymbol{r}_{t'} = \boldsymbol{r}^0$. The second statement in Theorem 6.2 states that, once we reach the recurrent configuration of the oil game, we have a constant oil flow $\varphi(v) = 1$ through each edge. In particular, for all $t' > 3\,n\,\|\boldsymbol{r}_0\|^-/(\lambda+1)$, we have that $\Phi_{t'}(v) > 0$ for all $v \in V(G)$. Because of Lemma 7.1, this means that $\Psi_{t''}(v) > 0$ for all $v \in V(G)$ and all integers $t'' > t^s + 3\,n\,\|\boldsymbol{r}_0\|^-/(\lambda+1)$. But if $\Psi_{t''}(v) > 0$ for every $v \in V(G)$, then every vertex must have fired at least once. Because of Corollary 3.7, we can conclude that $\boldsymbol{s}_{t''}$ is a critical configuration. From part (a) we know

$$t^s \leqslant \left\lceil \frac{3\,n\,\|\boldsymbol{s}_0\|^+}{\lambda+1} \right\rceil < \frac{3\,n\,\|\boldsymbol{s}_0\|^+}{\lambda+1} + 1.$$

We also know that, if $s_0(v) < 0$, then $s_0(v) \leqslant s_{t^s}(v) \leqslant d_G(v) - 1$, hence $s_0(v) - d_G(v) + 1 \leqslant r_0(v) \leqslant 0$; and if $s_0(v) \geqslant 0$, then $s_{t^s}(v) \geqslant 0$, hence $-d_G(v) + 1 \leqslant r_0(v)$. We find

$$\|\boldsymbol{r}_0\|^- \leqslant \|\boldsymbol{s}_0\|^- + \sum_{v \in V(G)\setminus\{q\}} d_G(v) < \|\boldsymbol{s}_0\|^- + 2\,m.$$

Combining everything, this gives that, for all integers t with

$$t > \frac{3\,n\,\|\boldsymbol{s}_0\|^+}{\lambda+1} + 1 + \frac{3\,n\,(\|\boldsymbol{s}_0\|^- + 2\,m)}{\lambda+1} = \frac{3\,n\,(\|\boldsymbol{s}_0\| + 2\,m)}{\lambda+1} + 1,$$

$\boldsymbol{s}_t$ is a critical configuration. □

Corollary 7.4. *Let $\boldsymbol{s}_0$ be a starting configuration of the dollar game on the graph G with n vertices and edge-connectivity λ.*

(a) *If $\boldsymbol{s}_0$ is not stable, then, after at most $3\,n^2\,\|\boldsymbol{s}_0\|^+/(\lambda+1)$ firings, involving at most $6\,n\,m\,\|\boldsymbol{s}_0\|^+/(\lambda+1)$ chip movements, a stable configuration is obtained.*

(b) *And, after at most $3\,n^2\,(\|\boldsymbol{s}_0\| + 2\,m)/(\lambda+1) + n$ firings, involving at most $6\,n\,m\,(\|\boldsymbol{s}_0\| + 2\,m)/(\lambda+1) + 2\,m$ chip movements, a critical configuration is obtained.*

Proof. This follows directly by combining Theorem 7.3 with Lemma 4.2(b) and (c). □

We can now prove most of the theorems in Section 1. Theorem 1.5 follows directly from Corollary 7.4(b).

Proof of Theorem 1.6. Using Corollary 3.5, the vector sum $\boldsymbol{c}_1 + \boldsymbol{c}_2$ of two critical configurations $\boldsymbol{c}_1, \boldsymbol{c}_2$ satisfies $0 \leqslant (c_1 + c_2)(v) \leqslant 2\,d_G(v) - 2$ for all $v \in V(G)\setminus\{q\}$. This gives

$$\|\boldsymbol{c}_1 + \boldsymbol{c}_2\| < 2 \sum_{v \in V(G)\setminus\{q\}} d_G(v) \leqslant 4\,m.$$

From Corollary 7.4(a) we see that this means that, after at most $O(n^2\,m\,\lambda^{-1})$ firings, involving at most $O(n\,m^2\,\lambda^{-1})$ chip movements, a stable configuration is obtained. By Corollary 3.9 this stable configuration is in fact equal to $\boldsymbol{c}_1 \bullet \boldsymbol{c}_2$. □

8. Another way to obtain a critical configuration

In this section we show how, starting with any configuration $\boldsymbol{s}_0$ of the dollar game on a graph G, we can find a stable configuration $\boldsymbol{s}$ such that $\gamma(\boldsymbol{s}) = \gamma(\boldsymbol{s}_0)$ and $\|\boldsymbol{s}\| \leqslant 2\,m$.

Following Corollary 7.4(b), we see that we need at most $12\,n^2\,m/(\lambda+1)+n$ firings, involving at most $24\,n\,m^2/(\lambda+1)+2\,m$ chip movements, to find $\gamma(\boldsymbol{s})$ for such an $\boldsymbol{s}$. The crucial fact is that this configuration $\boldsymbol{s}$ can be found very rapidly, provided we allow ourselves to perform arithmetic operations involving rationals with numerator and denominator of the order $\max\{\,\max_{v\in V(G)\setminus\{q\}}|s(v)|,\,\kappa\}$.

The approach in this section is based on a more algebraic way to look at configurations of the dollar game. Let the *Laplacian matrix* Q of G be the $n\times n$ matrix with

$$Q(u,v) \;=\; \begin{cases} -e(u,v), & \text{if } u\neq v,\\ d_G(u), & \text{if } u=v, \end{cases} \qquad \text{for all } u,v\in V(G).$$

Let Q_q denote the matrix obtained from Q by deleting the row and column corresponding to the vertex q, and let Q_q^+ denote the matrix obtained from Q by only deleting the row corresponding to q. It is well known (see, *e.g.*, [3]) that the determinant of Q_q is equal to the tree number κ. For a vertex $v\in V(G)$, let $\boldsymbol{c}_q(v)$ be the column of Q_q^+ corresponding to v, which is the same column in Q_q for $v\neq q$. It is easy to show that

$$\boldsymbol{c}_q(q) \;=\; -\sum_{v\in V(G)\setminus\{q\}} \boldsymbol{c}_q(v). \tag{8.1}$$

The following lemma follows directly by comparing the definition of firing a vertex with the definition of Q. Also recall the definition of a legal sequence and a representative vector from the first two sections.

Lemma 8.1.

(a) *For any vertex $v\in V(G)$, firing vertex v when we are in a configuration $\boldsymbol{s}$ results in the configuration $\boldsymbol{s}-\boldsymbol{c}_q(v)$.*

(b) *Let $\boldsymbol{s}$ be a configuration, $\sigma=v_1,\dots,v_k$ a finite sequence of vertices which is legal for $\boldsymbol{s}$, and $\boldsymbol{x}_\sigma$ the representative vector of σ. Suppose that after applying σ to $\boldsymbol{s}$ we obtain the configuration $\boldsymbol{s}'$. Then*

$$\boldsymbol{s}' \;=\; \boldsymbol{s}-Q_q^+\,\boldsymbol{x}_\sigma \;=\; \boldsymbol{s}+\sum_{v\in V(G)\setminus\{q\}} (x_\sigma(q)-x_\sigma(v))\,\boldsymbol{c}_q(v). \tag{8.2}$$

Let $\mathbb{Z}^{n-1}$ be the $(n-1)$-dimensional vector-space over the integers, with coefficients indexed by elements of $V(G)\setminus\{q\}$. And let C_z be the subspace of $\mathbb{Z}^{n-1}$ spanned by all integer linear combinations of the columns of Q_q. It can be shown that if G is a connected graph, then the matrix Q_q is nonsingular and hence C_z has dimension $n-1$. Since $\mathbb{Z}^{n-1}$ with normal vector addition has the structure of an abelian group, and C_z is a subgroup of this group, we also have that the quotient $\mathbb{Z}^{n-1}/C_z$ is an abelian group. Since C_z is of full dimension, this quotient group is finite. It is called the *Picard group* in [4, 5].

As normal, for an $\boldsymbol{a}\in\mathbb{Z}^{n-1}$ the coset of $\boldsymbol{a}$ in $\mathbb{Z}^{n-1}/C_z$ will be denoted by $[\boldsymbol{a}]$. It follows from Lemma 8.1(b) that, if $\boldsymbol{s}$ is a configuration with legal sequence σ, and $\boldsymbol{s}'$ is the configuration obtained after σ has been applied, then $[\boldsymbol{s}']=[\boldsymbol{s}]$. This gives that, for any $\boldsymbol{s}\in\mathbb{Z}^{n-1}$, $[\boldsymbol{s}]=[\gamma(\boldsymbol{s})]$.

The following result, which immediately gives Theorem 1.2, is proved in [4, 5]. Note that the observation in the previous paragraph establishes the surjectivity of ξ.

Theorem 8.2 (Biggs [4, 5]). *The function* $\xi : K \longrightarrow \mathbb{Z}^{n-1}/C_z$ *defined by* $\xi(\boldsymbol{c}) = [\boldsymbol{c}]$ *is a bijection.*

It follows from Theorem 8.2 and the definition of C_z as the subspace formed by integer linear combinations of the columns of Q_q that, for any configuration $\boldsymbol{s}$ and $\boldsymbol{z} \in \mathbb{Z}^{n-1}$, if we set $\boldsymbol{s}' = \boldsymbol{s} + Q_q \boldsymbol{z}$, then $\gamma(\boldsymbol{s}') = \gamma(\boldsymbol{s})$.

For a real number a, let $\lfloor a \rfloor$ be the *floor of* a, that is, the largest integer smaller than or equal to a; and for a vector $\boldsymbol{a}$, let $\lfloor \boldsymbol{a} \rfloor$ be the integral vector obtained by taking the floor of every coefficient of $\boldsymbol{a}$. For a configuration of the dollar game $\boldsymbol{s}$ on a graph G define

$$\bar{\boldsymbol{s}} = \boldsymbol{s} - Q_q \lfloor Q_q^{-1} \boldsymbol{s} \rfloor.$$

Since $\lfloor Q_q^{-1} \boldsymbol{s} \rfloor \in \mathbb{Z}^{n-1}$, we have that $\gamma(\bar{\boldsymbol{s}}) = \gamma(\boldsymbol{s})$.

Lemma 8.3. *For any configuration* $\boldsymbol{s}$ *we have that* $-d_G(v) + 1 \leqslant \bar{s}(v) \leqslant d_G(v) - 1$ *for all* $v \in V(G) \setminus \{q\}$.

Proof. First set $\boldsymbol{t} = Q_q^{-1}\boldsymbol{s} - \lfloor Q_q^{-1} \boldsymbol{s} \rfloor$, hence $0 \leqslant t(v) < 1$ for all $v \in V(G) \setminus \{q\}$. We find that

$$\bar{\boldsymbol{s}} = \boldsymbol{s} - Q_q \lfloor Q_q^{-1} \boldsymbol{s} \rfloor = Q_q(Q_q^{-1}\boldsymbol{s} - \lfloor Q_q^{-1} \boldsymbol{s} \rfloor) = Q_q \boldsymbol{t}.$$

It follows that, for all $v \in V(G) \setminus \{q\}$,

$$\bar{s}(v) = (Q_q \boldsymbol{t})(v) = t(v)\, d_G(v) - \sum_{u \in N_G(v) \setminus \{q\}} t(u)\, e(u, v).$$

Since $0 \leqslant t(u) < 1$ for all $u \in V(G) \setminus \{q\}$, we also have that

$$0 \leqslant t(v)\, d_G(v) < d_G(v) \qquad \text{and} \qquad 0 \leqslant \sum_{u \in N_G(v) \setminus \{q\}} t(u)\, e(u, v) < d_G(v).$$

This gives that $-d_G(v) < \bar{s}(v) < d_G(v)$ for all $v \in V(G) \setminus \{q\}$. The result follows from the fact that $\bar{\boldsymbol{s}}$ is an integral vector. $\square$

We are now ready to give the proof of Theorem 1.7.

Proof of Theorem 1.6. Let $\boldsymbol{s}$ be any configuration of the dollar game on the graph G. Set $\boldsymbol{t} = Q_q^{-1}\boldsymbol{s} - \lfloor Q_q^{-1} \boldsymbol{s} \rfloor$, hence $\bar{\boldsymbol{s}} = Q_q \boldsymbol{t}$. We also have $\gamma(\bar{\boldsymbol{s}}) = \gamma(\boldsymbol{s})$.

In order to determine $\boldsymbol{t}$, we first calculate $Q_q^{-1}\boldsymbol{s}$. This is the product of an $(n-1) \times (n-1)$ matrix with an $(n-1)$-vector, which involves $O(n^2)$ operations. The entries of Q_q^{-1} are rational numbers with numerator and denominator of the order $\det(Q_q)$. Since, as mentioned before, $\det(Q_q) = \kappa$, the number of spanning trees of G, we find that to determine $Q_q^{-1}\boldsymbol{s}$ we need to do $O(n^2)$ arithmetic operations involving rational numbers with numerator and denominator of the order $\max\{\kappa, \max_{v \in V(G) \setminus \{q\}} |s(v)|\}$.

Once $Q_q^{-1}\boldsymbol{s}$ is calculated, $\boldsymbol{t} = Q_q^{-1}\boldsymbol{s} - \lfloor Q_q^{-1}\boldsymbol{s}\rfloor$ can be found in $O(n)$ arithmetic operations. And then to determine $\bar{\boldsymbol{s}} = Q_q\,\boldsymbol{t}$ takes $O(n^2)$ operations, although this time, because of Lemma 8.3, we are working with integers of order $\max_{v\in V(G)\setminus\{q\}} d_G(v)$. So in total we see that $\bar{\boldsymbol{s}}$ can be found using $O(n^2)$ arithmetic operations.

Once $\bar{\boldsymbol{s}}$ has been found, we start the normal chip-firing operations to find $\gamma(\bar{\boldsymbol{s}})$. By Lemma 8.3 we have that

$$\|\bar{\boldsymbol{s}}\| = \sum_{v\in V(G)\setminus\{q\}} |\bar{s}(v)| < \sum_{v\in V(G)} d_G(v) = 2\,m.$$

Because of Corollary 7.4(b) we find that $\gamma(\bar{\boldsymbol{s}})$ will be reached after $O(n^2\,m\,\lambda^{-1})$ firings, involving at most $O(n\,m^2\,\lambda^{-1})$ chip movements. This completes the proof of the theorem. □

Remark. The procedure described in the proof above assumes that the inverse matrix Q_q^{-1} has been determined before we start the calculations. The reason for this assumption is that the same matrix Q_q^{-1} is used every time $\gamma(\boldsymbol{s})$ must be found for a configuration $\boldsymbol{s}$; hence it seems reasonable to determine and store Q_q^{-1} when $\gamma(\boldsymbol{s})$ is needed for many initial configurations $\boldsymbol{s}$.

If we cannot assume that the inverse matrix Q_q^{-1} is known beforehand, then we can find $Q_q^{-1}\boldsymbol{s}$ by solving the system of linear equations $Q_q\,\boldsymbol{x} = \boldsymbol{s}$. Using straightforward Gaussian elimination, this will involve $O(n^3)$ arithmetic operations (in fact it is possible to do it more rapidly), instead of the $O(n^2)$ mentioned in Theorem 1.7.

References

[1] Bak, P. (1996) *How Nature Works: The Science of Self-Organised Criticality*, Springer.

[2] Bak, P., Tang, C. and Weisenfeld, K. (1988) Self-organised criticality. *Phys. Rev. A* **38** 364–374.

[3] Biggs, N. (1993) *Algebraic Graph Theory*, 2nd edn, Cambridge University Press.

[4] Biggs, N. L. (1999) Chip firing and the critical group of a graph. *J. Alg. Combin.* **9** 25–45.

[5] Biggs, N. (1997) Algebraic potential theory on graphs. *Bull. London Math. Soc.* **29** 641–682.

[6] Biggs, N. (1999) The Tutte polynomial as a growth function. *J. Alg. Combin.* **10** 115–133.

[7] Biggs, N. and Winkler, P. (1997) Chip-firing and the chromatic polynomial. Preprint, London School of Economics.

[8] Björner, A. and Lovász, L. (1992) Chip-firing games on directed graphs. *J. Alg. Combin.* **1** 305–328.

[9] Björner, A., Lovász, L. and Schor, P. W. (1991) Chip-firing games on graphs. *Europ. J. Combin.* **12** 283–291.

[10] Bollobás, B. (1998) *Modern Graph Theory*, Springer.

[11] Bondy, J. A. and Murty, U. S. R. (1976) *Graph Theory with Applications*, Macmillan and Elsevier.

[12] Coppersmith, D., Feige, U. and Shearer, J. (1996) Random walks on regular and irregular graphs. *SIAM J. Discrete Math.* **9** 301–308.

[13] Dhar, D. (1990) Self-organised critical state of sandpile automation models. *Phys. Rev. Lett.* **64** 1613–1616.

[14] Dhar, D., Ruelle, P., Sen, S. and Verma, D. (1995) Algebraic aspects of abelian sandpile models. *J. Phys. A* **28** 805–831.

[15] Eriksson, K. (1991) No polynomial bound for the chip firing game on directed graphs. *Proc. Amer. Math. Soc.* **112** 1203–1205.

[16] Grötschel, M., Lovász, L. and Schrijver, A. (1993) *Geometric Algorithms and Combinatorial Optimization*, 2nd corr. edn, Springer.
[17] van den Heuvel, J. Random generation of spanning trees involving chip-firing. In preparation.
[18] Lovász, L. and Winkler, P. (1995) Mixing of random walks and other diffusions on a graph. In *Surveys in Combinatorics, 1995* (P. Rowlinson, ed.), Cambridge University Press, pp. 119–154.
[19] Merino Lopez, C. (1997) Chip-firing and the Tutte polynomial. *Ann. Combin.* **1** 253–259.
[20] Moore, C. and Nilsson, M. (1999) The computational complexity of sandpiles. *J. Statist. Phys.* **96** 205–224.
[21] Rossin, D. (1999) On the complexity of addition in the sandpile group of a graph. Preprint.
[22] Tardos, G. (1988) Polynomial bound for a chip firing game on graphs. *SIAM J. Discrete Math.* **1** 397–398.