

On the dynamics of Social Balance on general networks (with an application to XOR-SAT)

Gabriel Istrate

eAustria Research Institute,

Bd. V. Pârvan 4, cam. 045B,

Timișoara RO-300223, Romania.

Email: gabrielistrate@acm.org

Abstract. We study nondeterministic and probabilistic versions of a discrete dynamical system (due to T. Antál, P. L. Krapivsky, and S. Redner [3]) inspired by Heider’s social balance theory. We investigate the convergence time of this dynamics on several classes of graphs. Our contributions include:

1. We point out the connection between the triad dynamics and a generalization of *annihilating walks* to hypergraphs. In particular, this connection allows us to completely characterize the recurrent states in graphs where each edge belongs to at most two triangles.
2. We also solve the case of hypergraphs that do not contain edges consisting of one or two vertices.
3. We show that on the so-called “triadic cycle” graph, the convergence time is linear.
4. We obtain a cubic upper bound on the convergence time on 2-regular triangular simplexes G . This bound can be further improved to a quantity that depends on the *Cheeger constant* of G . In particular this provides some rigorous counterparts to experimental observations in [19].

We also point out an application to the analysis of the random walk algorithm on certain instances of the 3-XOR-SAT problem.

Keywords: social balance, discrete dynamical systems, rapidly mixing Markov chains, XOR-SAT.

1. Introduction

Heider’s balance theory [13] is a well-established subject in social psychology. It is a theory of cognitive consistency that postulates that the evolution of interpersonal relations in a group of individuals is dominated by a drive towards *psychological balance*. It is one of the first theories of social sciences that has benefited (among the first) from concepts and methods from graph theory [6].

Heider’s balance theory aimed to describe the “equilibrium” properties of the interpersonal relations, and did not include a dynamical component. Recently, however, dynamical models based on Heider’s balance theory have become popular in the statistical physics [16, 15] and social simulation literature [14, 23].

In this paper we study a particular model of this type, due to T. Antál, P. L. Krapivsky and S. Redner [3]. The dynamics (referred in the sequel as *the triadic dynamics*, and described more precisely in Section 2 below) is parametrized by a constant $p \in [0, 1]$. In [3] the authors investigated the dynamics on the complete graph K_n and displayed a double phase transition in the convergence time:

1. For $p < 1/2$ the convergence time is exponential.
2. For $p = 1/2$ the convergence time scales like $N^{4/3}$.
3. For $p > 1/2$ the convergence time is logarithmic in the system size n .

The triad dynamics was further investigated by means of computer simulations in [19]. The graph topology in this paper is a finite section of the triangular lattice. A phase transition is also displayed around a critical value $p_C \sim 0.4625$. In this case the convergence time is polynomial for $p < p_C$ and logarithmic for $p > p_C$.

A completely different reason for our interest in the triad dynamics is its somewhat unexpected relation [20] to *3-XOR-SAT*, a combinatorial problem that was investigated in Theoretical Computer Science [8] and Statistical Physics [21]. In particular, using the methods we develop for the analysis of the dynamics we will upper bound the expected convergence time of a simple local search algorithm, called *RandomWalk*, previously analyzed for random 3-XOR-SAT in [22, 4]. Unlike these papers, in our cases the formulas we will analyze our algorithm will be drawn from the so-called “planted assignment” model. The convergence time will depend critically (Theorem 7.1 below) on the structure of the input formula.

2. Preliminaries

The following is a formal definition of the dynamics we will be investigating in this paper:

Definition 2.1. Nondeterministic Triadic Dynamics. We start with a graph $G = (V, E)$ whose edges are labeled ± 1 . A triangle T in G is called *balanced* if the product of the labels of its edges is equal to 1. At any step t , for any imbalanced triangle T we are allowed to change the sign of an arbitrary edge of T (thus making T balanced). The move might, however, make other triangles unbalanced.

Definition 2.2. Probabilistic Triadic Dynamics. A probabilistic version of the dynamics in Definition 2.1, parametrized by a real number $p \in (0, 1)$, is specified [3] as follows: While there exists an imbalanced triangle, first choose uniformly at random an imbalanced triangle T . If the triangle T has a single negative edge e :

- With probability p turn e to positive.
- With probability $1 - p$ turn one of the other two edges of T to negative.

otherwise (T has three imbalanced edges) change the label of a random edge of T .

Definition 2.3. A *triadic simplicial complex* is a graph $G = (V, E)$ such that all edges $e \in E$ are part of some triangle of G .

Definition 2.4. The *triadic dual* of graph G is an undirected hypergraph with self-loops $T_3(G) = (\overline{V}_3, \overline{E}_3)$ defined as follows: \overline{V}_3 is the set of triangles of G . Hyperedges in $T_3(G)$ correspond to edges in G and connect all vertices $v \in \overline{V}_3$ containing a given edge. In particular we add a self-loop to vertex $v \in \overline{V}_3$ if v belongs to a unique triangle of G . We may even add two self-loops to the same vertex T if two of its edges belong only to triangle T .

We will also require that the triadic dual of graph G be connected. If this was not the case, the dynamics would decompose on independent dynamics on the connected components.

Definition 2.5. The *triadic cycle* TC_n is the graph G consisting of n triangles chained together. Edges AB and CD of the extreme triangles in the chain are “glued” by identifying $A \equiv C, B \equiv D$.

The construction is illustrated in Figure 1, where the triadic cycle graph with sixteen triangles is displayed, together with its triadic dual (in this graph circles represent self-loops).

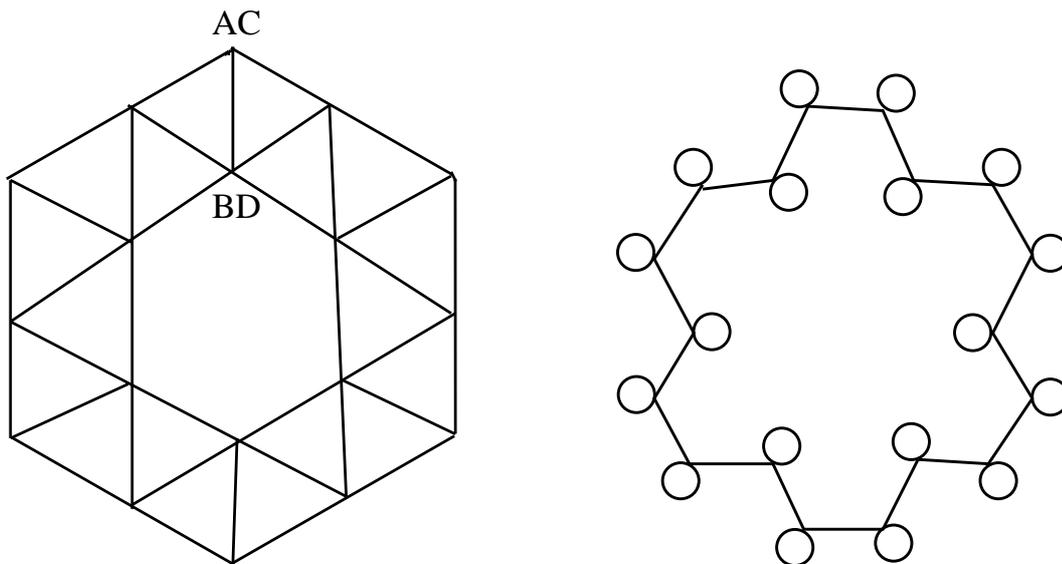


Figure 1. (a). The triadic cycle TC_{16} . (b). Its triadic dual

Definition 2.6. Let $G = (V, E)$ be a graph. For $S \subseteq V$ denote by $vol(S) = \sum_{x \in S} d(x)$ and $E(S, \overline{S})$ the set of edges connecting a node in S to one in \overline{S} .

The *Cheeger time of graph G* is

$$\tau_c(G) = \sup_{A \subseteq V} \frac{\pi[A] \cdot \pi[\overline{A}]}{\sum_{i \in A} \sum_{j \in \overline{A}} \pi[i] p_{i,j}},$$

where $\pi[A] = \frac{|A|}{n}$ and $p_{i,j} = 1/deg(i)$ iff $i \sim j$, 0 otherwise.

Remark 2.1. When G is r -regular $\tau_c(G) = \sup_{A \subseteq V} \frac{|A| \cdot |\bar{A}| / n^2}{r/n \cdot |E(A, \bar{A})|} = \sup_{A \subseteq V} \frac{|A| \cdot |\bar{A}|}{rn \cdot |E(A, \bar{A})|}$.

Finally, we need the following concept from [1]:

Definition 2.7. Let $(X_n), (Y_n)$ be independent random walks on G . The worst case meeting time of G is defined as:

$$\tau_M(G) = \max_{i,j} E[T_M | X_0 = i, Y_0 = j],$$

where $T_M = \min\{k : X_k = Y_k\}$.

3. Connection with annihilating walks and XOR-SAT

The triad dynamics can be mapped to a generalization of *annihilating walks*.

Definition 3.1. An *annihilating walk (AW)* on graph G is a nondeterministic dynamical system described as follows: Start with a ball at some of the vertices of the random graph. At each step we move one ball to a neighbor of its current vertex. When two balls meet at a vertex v they annihilate each other, and are subsequently eliminated from graph G .

The stochastic version of annihilating walks (called *annihilating random walks* [10]) have been studied in the interacting particle systems literature (e.g. [11]). A few recent results [7],[2] deal with interacting particle systems on a finite graph as well.

We will also need the following:

Definition 3.2. A *coalescing random walk (CRW)* on graph G is a stochastic process described as follows:

1. Start with a ball at some of the vertices of the random graph.
2. Each ball performs a random walk on G .
3. When two or more balls meet at a vertex v they coalesce, performing a common random walk on G .

Definition 3.3. For an annihilating random walk we denote by $T_{ARW}(G)$ the random variable defined as the number of steps until the number of balls on graph G becomes zero (one if the initial number of balls was odd). For a coalescing random walk we denote by $T_{CRW}(G)$ the random variable defined as the number of steps until all the balls coalesce.

To recover the connection between annihilating walks and triad dynamics, we have to define a generalization of annihilating walks to hypergraphs.

Definition 3.4. The *hyperedge switching process (HS)* on a hypergraph G is a nondeterministic dynamical system described as follows: Start with a ball at some of the vertices of the hypergraph. At each step we choose a hyperedge C in G that contains at least a ball. Any self-loop on a vertex that contains a ball counts as one of these such cliques. We then put balls on all empty nodes of C and remove the balls from the nonempty nodes.

Algorithm RandomWalkSat(Φ):

```

Start with an arbitrary assignment  $U$ .
while (there exists some unsatisfied clause)
    pick a random unsatisfied equation  $C$ 
    change the value of a random variable of  $C$  in  $U$ 
return assignment  $U$ .

```

Figure 2. The RandomWalkSat algorithm

Lemma 3.1. The nondeterministic triad dynamics on graph G and initial configuration s corresponds to the hyperedge switching process on graph $T_3(G)$ to the initial configuration \bar{s} defined by putting a ball to all vertices $v \in T_3(G)$ corresponding to unbalanced triangles in G .

Our problem also displays an unexpected connection with a problem in the area of satisfiability solving: *random 3-XOR SAT*. The satisfiability of random instances of this problem has been investigated in both Statistical Physics [21] and Theoretical Computer Science [8].

Definition 3.5. An instance F of the 3-XOR SAT problem is specified by a list of m equations on n variables, $x_{i_1} \oplus x_{i_2} \oplus x_{i_3} = b_i$, for some $b_i \in \{0, 1\}$ and $i_1, i_2, i_3 \in 1 \dots n$. F is *satisfiable* if there exists an assignment A of variables in F that makes every equation evaluate to true.

Our results will not involve random instances, but satisfiable instances from *the planted assignment model*. An instance of this problem is obtained by first choosing a random satisfying assignment A and then generating an instance compatible with A by only considering equations satisfied by A . We will also assume that instances we work with are *reduced*, i.e. they satisfy the following two conditions: no variable appears in a single clause and no two clauses share two variables. It is easy to see that any formula can be transformed to a reduced one. We simply eliminate clauses involving pure variables (since they can always be satisfied by setting the pure variable the right way). Also, if C_1 and C_2 share two variables, say x and y , let z and t the remaining variables. The conjunction of C_1 and C_2 entails a constraint of the type $z \oplus t = \lambda$, for some $\lambda \in \{0, 1\}$. We can thus eliminate one of the variables z, t (by replacing it with $t \oplus \lambda$). This eliminates one of the clauses C_1 and C_2 from the formula.

We will analyze the RandomWalk algorithm for 3-XOR SAT, displayed in Figure 2. We will assume that the input to the RandomWalk algorithm is a reduced formula and obtain upper bounds on the runtime of the algorithm for a class of formulas.

4. Recurrent states for nondeterministic triad dynamics

Definition 4.1. A state s is specified by giving a ± 1 label to each edge of graph G . A state is *recurrent* if for any infinite path $\pi = (\pi_0, \dots, \pi_n \dots)$ of the system s is reachable from π_n .

It is fairly easy to determine recurrent states if every edge is part of at most two triangles (that is, if the triadic dual is a graph with loops):

Theorem 4.1. Let G be a graph such that (i) the triangle graph of G is connected, and (ii) every edge of G is part of at most two triangles, and let $s_0 : E(G) \rightarrow \{\pm 1\}$ be an initial state. Then:

1. If G contains an edge that is part of only one triangle, then the recurrent states for the triad dynamics are exactly the *completely balanced states* reachable from s_0 .
2. If every edge in G is part of exactly two triangles and the number of imbalanced triangles in s_0 is even then the recurrent states for the triad dynamics are exactly the *completely balanced states* reachable from s_0 .
3. If every edge in G is part of exactly two triangles and the number of imbalanced triangles in s_0 is odd, then the recurrent states for the triad dynamics are the states containing *exactly one* imbalanced triangle.

Proof:

Define function P that maps each state s to the number of imbalanced triangles in state s .

Lemma 4.1. Function P is nondecreasing. That is, if $s \rightarrow t$ is a legal transition of the dynamics, then $P(s) \geq P(t)$.

Proof:

Consider the edge $e \in G$ chosen by the dynamics. There are three cases:

1. There is only one (imbalanced) triangle T that contains e . Then, by flipping the sign of edge e triangle T becomes balanced and no other triangle is affected. The number of imbalanced triangles goes down by one.
2. There are two triangles T_1, T_2 containing edge e (Figure 4), both imbalanced. Then, by flipping the sign of edge e triangles T_1, T_2 become balanced and the balancedness of no other triangle is affected. Hence the number of imbalanced triangles goes down by two.
3. There are two triangles T_1, T_2 containing edge e , one imbalanced (say it is T_1) and one (T_2) balanced. Then, by flipping the sign of edge e triangle T_1 becomes balanced, while T_2 becomes imbalanced. The balancedness of no other triangle is affected. Thus the number of unbalanced triangles stays the same.

□

Consider now a recurrent state t . It follows that there is no state w reachable from t with $P(w) < P(t)$.

To prove 1 note that from any state containing some imbalanced triangle we can reach a state w with one less balanced triangle by “propagating the imbalance” towards the triangle T with an edge e occurring only in T . In this process the number of imbalanced triangles stays the same. By then changing the label of e we decrease the number of imbalanced triangles.

The arguments for points 2 and 3 are similar, noting that the number of imbalanced triangles stays the same or decreases by exactly 2. \square

We would like to extend our result to general graphs G . However, this is an open problem so far. The reason is that for general graphs G the triadic dual is no longer a graph, but a hypergraph and the potential function is no longer nonincreasing. In the sequel we provide a partial result.

4.1. Recurrent configurations for the nondeterministic hyperedge switching process

In this section we study the hyperedge switching process on hypergraphs H whose edges have more than two vertices. We will refer to this condition as H does not contain graph edges.

Definition 4.2. Define, for each hyperedge e a variable z_e with values in \mathbf{Z}_2 . For any vertex $v \in V(H)$ we define the equation

$$\sum_{v \in e} z_e = w_2(v) - w_1(v). \quad (1)$$

We allow empty sums on the left side, and will denote the system by $H(w_1, w_2)$.

Definition 4.3. If x is a state on H and l is an edge of H , define

$$x^{(l)}(v) = \begin{cases} 1 + x(v), & \text{if } v \in l, \\ x(v), & \text{otherwise.} \end{cases} \quad (2)$$

In this section we prove the following:

Theorem 4.2. Let H be a connected hypergraph with no graph edges, let $w_1 \neq \mathbf{0}$ be an initial configuration and let w_2 be a final configuration.

There is a polynomial time algorithm to test whether w_2 is a recurrent state for the nondeterministic hyperedge process on H with starting state w_1 .

Proof:

Lemma 4.2. If state w_2 is reachable from w_1 then the system of equations $H(w_1, w_2)$ has a solution in \mathbf{Z}_2 .

Proof:

Let P be a path from w_1 to w_2 and let z_e be the number of times edge e is used on path P (mod 2). Then $(z_e)_{e \in E}$ is a solution of system (1). Indeed, element $w(v)$ (viewed modulo 2) flips its value anytime an edge containing v is scheduled. \square

Lemma 4.2 gives a necessary condition for reachability in Theorem 4.2. The complete characterization of recurrent states is a consequence of the following two lemmas:

Lemma 4.3. Let H be a hypergraph with no graph edges (not necessarily connected). Assume that $w_1, w_2 \in \{0, 1\}^{V(H)}$ are configurations such that for no connected component C of H , $w_1|_C \equiv \mathbf{0}$, $w_2|_C \not\equiv \mathbf{0}$. Then state w_2 is reachable from w_1 **if and only if** system of equations $H(w_1, w_2)$ has a solution in \mathbf{Z}_2 .

Proof:

We prove the result by induction on the number of edges in hypergraph H .

1. **Case $m = 1$:** Suppose system $H(w_1, w_2)$ has a solution. Since H contains a single edge e , $w_2(v) = w_1(v)$ for all vertices $v \notin e$ (otherwise the system would contain equation $0 = 1$). Also, for all $v \in e$ the quantity $w_2(v) - w_1(v) := \lambda$ does not depend on v (otherwise we would have two contradicting equations). There are two cases: $\lambda = 0$, in which case $w_1 = w_2$ and path P can be taken as the empty path, and the case $\lambda = 1$. In this case it follows that edge e contains at least one vertex v with $w_1(v) = 1$. Indeed, if this was not true then $w_1|_e \equiv \mathbf{0}$, and $w_2|_e \equiv \mathbf{1}$, contradicting the restriction from the hypothesis. One can then set $P = \{e\}$ and obtain a path from w_1 to w_2 .

2. **Case $m \geq 2$:**

Assume Lemma 4.3 is true for all hypergraphs with less than m edges, and consider an arbitrary hypergraph H with m edges. Without loss of generality we may assume that H is connected, otherwise we find paths separately on every connected component. There are two cases:

(a) **For some $l \in E$ system $H(w_1, w_2)$ has a solution with $z_l = 0$.**

Then the system $U(w_1, w_2)$, corresponding to hypergraph $U = H \setminus \{l\}$ is solvable; indeed every solution of $H(w_1, w_2)$ with $z_l = 0$ is also a solution of $U(w_1, w_2)$.

Hypergraph U has four types of connected components:

- i. Connected components P of U such that $w_1|_{P \setminus \{l\}} \neq \mathbf{0}$.
- ii. Connected components Q of U such that $w_1|_Q \equiv \mathbf{0}$ but there exists $v \in l \cap Q$ with $w_1(v) = 1$.
- iii. Connected components R of U such that $w_1|_R \equiv \mathbf{0}$, $w_2|_R \neq \mathbf{0}$.
- iv. Connected components S of U such that $w_1|_S \equiv \mathbf{0}$, $w_2|_S \equiv \mathbf{0}$.

Since H was connected, all such components contain at least one vertex from l . Moreover, we can assume that there exist no components of type (iv), since we can eliminate them from consideration as they do not affect the overall result. Also, the induction step is trivial if U has no components of type (iii). So assume that U contains some component of type (iii).

- **Case 1: $w_1|_l \neq \mathbf{0}$ and there exists $z \in l$ such that either $w_1(z) = 0$ and z belongs to a component of type (i) or $w_2(z) = 0$ and z belongs to a component of type (iii).**

We update the state using the following sequence of steps.

- 1: Schedule edge l . This is possible since the current configuration contains, by the hypothesis, at least a node $v \in l$ with $w_1(v) = 1$. In turn, this changes the states of nodes in l that belong to components of type (iii) to one.
- 2: Schedule components of type (iii), moving the states of their nodes from $w_1^{(l)}$ to $w_2^{(l)}$. This is possible since systems $U(w_1, w_2)$ and $U(w_1^{(l)}, w_2^{(l)})$ are equivalent (thus satisfiable), and state $w_1^{(l)}$ contains a nonzero value on every connected component of type (iii).

- 3: Schedule edge l again. This is possible since the state of node z is 1. This makes the states of nodes t in l that belong to components of type (iii) take value $w_2(t)$. Also, this changes the values of other nodes $t \in l$ to $w_1(t)$ again.
- 4: Finally, schedule components of type (i) and (ii), changing the values of their nodes from w_1 to w_2 . This is done componentwise, using the induction hypothesis, and the nonzero values of w_1 in each such component.

• **Case 2: $w_1|_l \neq 0$, Case 1 doesn't apply and there exists P a component of type (i) or (ii) with $w_2(z) = 1$ for some $z \in l \cap P$.**

Since Case 1 does not apply, we can identify the values of w_1 on every vertex $x \in l$: $w_1(x) = 1$ if x belongs to a connected component of type (i) or (ii), $w_1(x) = 0$ otherwise. Also, if $x \in l$ belongs to a configuration of type (iii), $w_2(x) = 1$.

Consider an edge l_2 such that $l_2 \cap l \neq \emptyset$ and l_2 belongs to a component of type (iii). l_2 exists (i.e. Q does not consist of an isolated vertex in U) otherwise the component Q would yield an unsatisfiable equation $0 = 1$ in $U(w_1, w_2)$.

- 1: Schedule components of type (i) and (ii), changing the values of their nodes from w_1 to w_2 .
- 2: Schedule edge l . This is possible since the current configuration contains, by the hypothesis, at least a node $z \in l$ with $w_2(z) = 1$. In turn, this turns the states of nodes in l that belong to components of type (iii) to one.
- 3: Schedule all components P of type (iii) except Q , moving the states of their nodes from $w_1^{(l)}$ to $w_2^{(l)}$, restricted to the scheduled components. This is possible since systems $U(w_1, w_2)$ and $U(w_1^{(l)}, w_2^{(l)})$ are equivalent.
- 4: Schedule component Q , moving its state from $w_1^{(l)}|_Q$ to $(w_2^{(l)})|_Q$. This is possible since system $U(w_1^{(l)}, w_2^{(l)})$ is solvable.

Note that, since Case 1 doesn't apply, $w_2^{(l)}(x) = 0$ for all $x \in Q \cap l$.

- 5: Let r be a node of Q with $w_2(r) = 1$, closest to some vertex of edge l in distance. Consider the path D from r to a node in l . All edges other than the one containing r have only zero values. Schedule edges on path D from r towards l , one by one (Figure 3(a)). This “propagates” the value 1 towards the vertex in l . Note that all nodes on intermediate edges had initially zero values, so they have intermediate nodes on the path retain value 1. In particular, last edge has a node in $Q \setminus \{l\}$ whose value is one.

Also, this action turns the values of nodes in $l \cap Q$ to one.

- 6: Schedule edge l again. This restores the correct value of all nodes in $l \setminus Q$. It also turns nodes in $l \setminus Q$ to zero.
- 7: “Undo” scheduling nodes on D from l to r (Figure 3(b)). This is possible since, as each edge has ≥ 3 vertices, last edge had a node whose value is one, and “we propagate this value” towards r . Since edges of D were scheduled twice, every label is correct.

• **Case 3: $w_1|_l \neq 0$ and Cases 1 and 2 do not apply.**

Since Cases 1 and 2 do not apply, we can identify the values of w_1, w_2 on every vertex $x \in l$: $w_1(x) = 1, w_2(x) = 0$ if x belongs to a connected component of type (i)

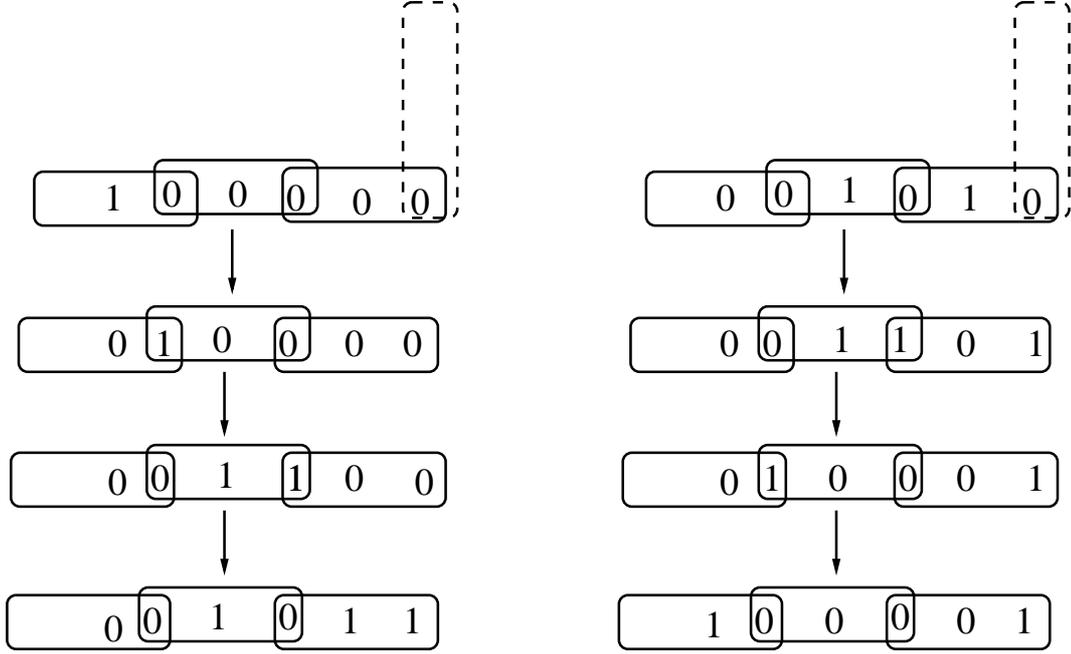


Figure 3. (a). Forward propagation of ones. (b). Backward propagation. In both cases edge l is shaded.

or (ii), $w_1(x) = 0$ otherwise. Also, if $x \in l$ belongs to a configuration of type (iii), $w_1(x) = 0, w_2(x) = 1$.

The construction is almost identical to that of Case 3. The only difference is that Step 1 is now executed at the very end of the process. This is possible, since before executing this step all vertices $x \in l$ belonging to components of type (i) or (ii) have value 1 (edge l being scheduled twice was the only action affecting their value).

- **Case 4:** $w_1|_l \equiv \mathbf{0}$. We reduce this case to one of the previous three cases as follows: since w_1 is not equal to $\mathbf{0}$, there exists a vertex v reachable from l in H with $w_1(v) = 1$. Choose such a vertex v at minimal distance from l . Let this path be D . First, use the forward propagation trick on D to change the state w_1 to a state w_3 with $w_3|_l \neq \mathbf{0}$. System $H(w_3, w_2) = 0$ has a solution with $z_l = 0$, since system $H(w_1, w_2) = 0$ has one. We then apply one of cases one to three, the one that works, to the former system.

(b) **For all edges l , all solutions of system $H(w_1, w_2) = 0$ have $z_l = 1$.**

If there exists l with $w_1|_l \neq \mathbf{0}$ and $w_1^{(l)} \neq \mathbf{0}$ then system $H(w_1^{(l)}, w_2) = 0$ has a solution with $z_l = 0$ (any solution of $H(w_1, w_2) = 0$ with the value of z_l flipped to zero). We then apply one of the previous cases.

Otherwise for all l with $w_1|_l \neq \mathbf{0}$, $w_1^{(l)} \equiv \mathbf{0}$. Therefore H has a single such edge, and the result immediately follows. □

Lemma 4.4. Let H be a connected uniform hypergraph, and let w_1, w_2, w_3 be configurations on H . If w_2 is reachable from w_1 and $w_3 \neq \mathbf{0}$ is reachable from w_1 in one step then w_2 is reachable from w_3 .

Consequently one of the following two alternatives hold:

- $\mathbf{0}$ is reachable from w_1 and is the only recurrent state for the process started at w_1 .
- $\mathbf{0}$ is not reachable from w_1 and w_2 is a recurrent state for the nondeterministic hyperedge switching process on H started at w_1 .

Proof:

If system $H(w_1, w_2)$ has a solution (z_e) and l is the edge flipped when going from w_1 to w_3 , let

$$\overline{z}_e = \begin{cases} z_e, & \text{if } e \neq l, \\ 1 + z_l, & \text{otherwise.} \end{cases} \quad (3)$$

It is easy to see that (\overline{z}_e) is a solution to system $H(w_3, w_2)$. Since w_2 is special, the restriction in Lemma 4.3 is satisfied for the pair (w_3, w_2) . \square

Applying Lemma 4.3 and Lemma 4.4 we obtain an easy algorithm for recurrence: make two reachability tests using Lemma 4.3. \square

5. Time to social balance on the triadic cycle

Theorem 4.1 motivates the study of the following quantity:

Definition 5.1. For a triangular simplicial complex G , denote by $\tau_{SB}(G)$ the maximum, over all initial states s , of $E[T_s(G)]$, where $T_s(G)$ is the expected time for the system, starting from initial state s to enter a state with zero (one) imbalanced triangles (according to the result in Theorem 4.1). $\tau_{SB}(G)$ will be called *the time to social balance* of graph G .

We first investigate the convergence time of random triad dynamics on the triadic cycle T_n . For this class of graphs, Theorem 4.1 states that all recurrent states are socially balanced. We were motivated in our particular choice of graph G by our paper [9], which investigated a similar dynamics. The dynamics in [9] can also be described using the particle analogy, and is specified by particle creation/annihilation rules $(1, 0) \rightarrow (1, 1)$, $(0, 1) \rightarrow (1, 1)$, $(1, 1) \rightarrow (0, 0)$. In contrast, the annihilating random walk corresponding to the dynamics in this paper is specified by a very similar set of rules $(0, 1) \rightarrow (1, 0)$, $(1, 0) \rightarrow (0, 1)$, $(1, 1) \rightarrow (0, 0)$.

The next result shows that the convergence time to a socially balanced state is similar to the analog result in [9]. The proof is simpler, though:

Theorem 5.1. Let $p_- = \min\{p, 1/2\}$. The time to social balance on the triadic cycle T_n is $O(\frac{n}{p_-})$.

Proof:

The result follows from the following:

Lemma 5.1. Function P decreases at each step with probability at least p_- .

Proof:

Consider three adjacent triangles in the graph TC_n , as displayed in Figure 5. Suppose that triangle B is unbalanced and is the one chosen by the dynamics. There are two cases:

1. The dynamics changes the value of edge x (the one that is not part of any other triangle). In this case no triangle but B is affected, and the number of unbalanced triangles goes down by one.
2. The dynamics changes the value of one of the edges y, z . In this case triangle T_2 becomes balanced. The triangle that shares the chosen edge with T_2 becomes unbalanced if it was previously balanced, and balanced if it was unbalanced as well. All in all, the number of unbalanced triangles does not increase.

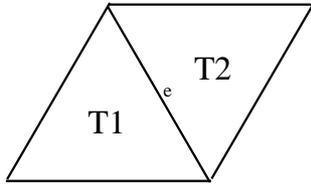


Figure 4. The two triangles in the proof of Lemma 4.1.

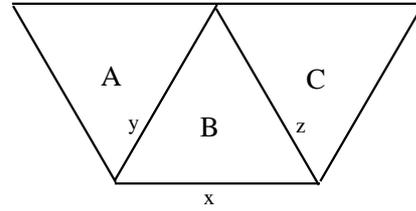


Figure 5. The three triangles in the proof of Lemma 5.1.

□

We now apply the following result ([18] Theorem 1.3 pp. 15):

Lemma 5.2. Let $g : \mathbf{R}_+ \rightarrow \mathbf{R}_+$ be a monotone nondecreasing function. Consider a particle whose position changes at integer moments and is always an integer. If the particle is at position $m > 1$, it proceeds at the next step to position $m - X$, where X is a random variable ranging over the integers $1, \dots, m - 1$. All we know about X is that $E[X] \geq g(m)$ and that X is chosen independently of the past.

Let T be the random variable denoting the number of steps in which the particle reaches position 1. Then $E[T] \leq \int_1^n \frac{dx}{g(x)}$.

Using this result we infer that $\tau_{SB}(G) \leq \frac{n}{p}$.

□

6. The case $p = 1/3$ on a 2-regular triangular simplex

Another case where the dynamics is easy to analyze is that when $p = 1/3$ and graph G is a 2-regular triangular simplex. In this case, according to Theorem 4.1, all recurrent states are either socially balanced (fixpoints), or all of them contain exactly one imbalanced triangle. Radicchi et al. [19] experimentally study the convergence time on the triangular lattice and obtain the estimate $\theta(N^\alpha)$, with $\alpha \sim 2.24$.

We can estimate the convergence time as follows:

Theorem 6.1. For any 2-regular triangular simplex G such that $T_3(G)$ is connected there exists $C > 0$ such that the time to social balance on G satisfies

$$\tau_M(T_3(G)) \leq \tau_{SB}(G) \leq T_{CRW}(T_3(G)) \leq C \cdot n^{\min\{3, 2 + \log_2(\tau_C(T_3(G)))\}}, \quad (4)$$

where $\tau_C(T_3(G))$ is the Cheeger constant of the triangular dual $T_3(G)$.

Before going into the proof, note that an open problem in [2] (Chapter 14, Open problem 13), combined with Proposition 5 (same chapter) would imply that the lower and upper bounds in Theorem 6.1 stated in terms of graph $T_3(G)$ have the same order of magnitude.

Proof:

Since G is 2-regular, $T_3(G)$ is a 3-regular *graph*, has no loops, and the clique switching process reduces to an annihilating random walk.

The annihilation time of an ARW is stochastically dominated by the coalescence time of a CRW on the same graph. This is an easy example of a coupling [2]:

Lemma 6.1. We can couple the annihilating and coalescing random walk such that $T_{ARW} \leq T_{CRW}$. Consequently

$$T_{ARW}(G) \leq T_{CRW}(G).$$

Proof:

The coupling uses a well known idea (see e.g. [11]). We first describe the process intuitively. Consider a coalescing random walk. Declare particles that coalesce to be “ghost” particles (as opposed to the “live”, not yet coalesced ones). If a set of ghost particle meet a real particle they stick to it, adopting the (randomly chosen) trajectory of the live particle. When two live particle meet they become ghosts. The coupling is obtained by seeing the ARW as the CRW restricted to live particles.

Formally, define $(\xi_t^x)_{x \in V, t \geq 0}$ to be a stochastic process such that

- $\xi_0^x = x$ for all $x \in V$.
- $\xi_t^x = \xi_t^y$ implies that $\xi_{t+s}^x = \xi_{t+s}^y$ for all $s \geq 0$.
- For all $x \in V$, $(\xi_t^x)_{t \geq 0}$ is a random walk on G .

Define the annihilating system $\eta_t \subseteq V$,

$$\eta_t = \{y \mid |\{x \in V, \xi_t^x = y\}| \text{ is odd}\}.$$

Assume that initially the number of particle was even. It is easy to see that when all particles have coalesced then all particles are ghost particles. The reason is simple:

- Any cluster of particle contains at most one live particle.
- The only way for a live particle to become a ghost is to meet another live particle.

- The number of live particle only goes down by two at a time.

□

For the upper bound we apply a result due to Aldous and Fill (Proposition 9, Section 3.14, Chapter 14 in [2]), generalizing an earlier result of Donnelly and Welsh [7]¹ and infer the theorem. We use the fact that the lattice graph is 3-edge-connected and 3-regular.

For the lower bound we consider all initial states consisting of exactly two imbalanced triangles, and use the connection with annihilating random walks.

□

7. Application to random 3-XOR-SAT

Definition 7.1. A formula is *connected* if one cannot partition F into two variable-disjoint formulas. F is *k-connected* if one can delete up to $k - 1$ variables (and the clauses of F where these variables appear) without disconnecting the formula.

As a consequence of our results we have:

Corollary 7.1. If reduced formula Φ has m equations, n variables, is 2-regular, and s -connected then the expected time until RandomWalk finds a solution satisfies:

$$ET_{RW} \leq \min\{m^3/2s, 2 \log 2 \cdot \tau_C(T_3(\Phi)) \cdot m^2\}.$$

Proof:

Consider the simplicial complex $S(\Phi)$ associated to the reduced formula and its triadic dual T_Φ . It is easy to see that the dynamics of The RandomWalk algorithm can be viewed as the annihilating random walk on T_Φ . We then apply the (discrete time version of) results of Donnelly and Welsh, as extended by Aldous and Fill in Chapter 14 in [2]. □

8. Conclusions

Our paper raises the interesting prospect that the convergence time of local search algorithms, such as the RandomWalk algorithm might be analyzed with ideas from the interacting particle systems literature. We believe that this would be especially interesting for “message-passing” algorithms such as the belief and *survey propagation* algorithms [5].

Clearly, we would like to see progress in analyzing the dynamics of the random hyperedge switching process on hypergraphs. In particular studying the mixing time of the random hypergraph switching process is a very interesting problem for further study. Another possible direction, motivated by the analogy with the dynamics studied in [9], is to further study the time to social balance using ideas similar to those in [17].

Finally, an interesting issue it to compute the Cheeger constant of the triangular lattice, as well as other planar regular lattices [12], and comparing the results with those in [19].

¹The result of Aldous and Fill is stated in [2] for continuous time, but can be easily translated to discrete time, at the expense of an additional linear factor

Acknowledgments

I thank Cornel Izbaşa and Cosmin Bonchiş for useful discussions.

This work has been supported by a Marie Curie International Reintegration Grant within the 6th European Community Framework Program.

References

- [1] Aldous, D.: Meeting times for independent Markov chains, *Stochastic Processes and their applications*, **38**, 1991, 185–193.
- [2] Aldous, D., Fill, A.: *Reversible Markov Chains and Random Walks on Graphs*, (manuscript in preparation), Available from <http://www.stat.berkeley.edu/~aldous/RWG/book.html>, 2007.
- [3] Antál, T., Krapivsky, P. L., Redner, S.: Social Balance on Networks: The Dynamics of Friendship and Enmity, *Physica D*, **224**(130), 2006.
- [4] Barthel, W., Hartmann, A., Weigt, M.: Solving satisfiability problems by fluctuations: The dynamics of stochastic local search algorithms, *Phys. Rev. E*, **67**(066104), 2003.
- [5] Braunstein, A., Mézard, M., Zecchina, R.: Survey propagation: an algorithm for satisfiability, *Random Structures and Algorithms*, **27**(2), 2005, 201–226.
- [6] Cartwright, D., Harary, F.: Structural balance: a generalization of Heider’s theory, *Psychological Review*, **63**, 1956, 277–293.
- [7] Donnelly, P., Welsh, D.: Finite particle systems and infection models, *Math.Proc. Cambridge Philos.Soc.*, **94**, 1983, 167–182.
- [8] Dubois, O., Mandler, J.: The 3-XORSAT threshold, *C. R. Math. Acad. Sci. Paris*, **335**(11), 2002, 963–966, ISSN 1631-073X.
- [9] Dyer, M., Greenhill, C., Goldberg, L., Istrate, G., Jerrum, M.: The Convergence of Iterated Prisoner’s Dilemma Game, *Combinatorics, Probability and Computing*, **11**, 2002, 135–147.
- [10] Erdős, P., Ney, P.: Some problems on random intervals and annihilating particles, *Annals of Probability*, **2**, 1974, 828–839.
- [11] Griffeath, D.: *Additive and Cancellative Interactive Particle Systems*, Springer Verlag, 1979.
- [12] Grünbaum, B., Shephard, G.: *Tilings and patterns*, W.H. Freeman & co., 1986.
- [13] Heider, F.: *The psychology of interpersonal relations*, John Wiley & Sons, 1958.
- [14] Hummon, N., Doreian, P.: Some dynamics of social balance processes: bringing Heider back into balance theory, *Social Networks*, **25**(1), 2003, 17–49.
- [15] Kulakowski, K.: Some Recent Attempts to Simulate the Heider Balance Problem, *Computing in Science and Engineering*, **9**(4), 2007, 80–85, ISSN 1521-9615.
- [16] Kulakowski, K., Gawronski, P., Gronek, P.: The Heider balance - a continuous approach, *International Journal of Modern Physics C*, **16**, 2005, 707.
- [17] Mossel, E., Roch, S.: Slow Emergence of Cooperation for Win-Stay Lose-Shift on Trees, *Machine Learning*, **7**(1–2), 2006, 7–22.
- [18] Motwani, R., Raghavan, P.: *Randomized Algorithms*, Cambridge University Press, 1995.

- [19] Radicchi, F., Vilone, D., Meyer-Ortmanns, H.: Universal Class of Triad Dynamics on a Triangular Lattice, *Physical Review E*, **021118**, 2006.
- [20] Radicchi, F., Vilone, D., Yoon, S., Meyer-Ortmanns, H.: Social balance as a satisfiability problem of computer science, *Physical Review E*, **026106**, 2006.
- [21] Ricci-Tersenghi, F., Weight, M., Zecchina, R.: Simplest random k -satisfiability problem, *Physical Reviews E*, **63**, 2001, 026702.
- [22] Semerjian, G., Monasson, R.: Relaxation and Metastability in a local search procedure for the random satisfiability problem, *Phys. Rev. E*, **67**(066103), 2003.
- [23] Wang, Z., Thorngate, W.: Sentiment and social mitosis: Implications of Heider's Balance Theory, *Journal of Artificial Societies and Social Simulation*, **6**(3), 2003.