Three Emergent Phenomena in the Multi-Turmite System and their Robustness to Asynchrony

Selma Belgacem, Nazim A. Fatès

Three Emergent Phenomena in the Multi-Turmite System and their Robustness to Asynchrony

Selma Belgacem and Nazim Fatès
INRIA Nancy - Grand Est / LORIA
615 rue du Jardin Botanique
54 600 Villers-lès-Nancy ; France

March 9, 2010

Abstract

The multi-turmite system is composed of concurrent Turing machines acting on a two-dimensional grid. The machines obey simple local rules, namely Langton's ant rules, their updating is considered under various simulation conditions: synchronous or asynchronous updating methods and different conflict resolution policies. We present three emergent phenomena : clocks, gliders and deadlocks. We study to which extent these phenomena are robust to changes in the updating and in the conflict resolution policy. Regularities of behaviour are observed from simulation results. We describe these regularities within the mathematical framework of discrete dynamical systems and show how their robustness can be analysed from a local "microscopic" view.

keywords concurrent Turing machines turmites ; Langton's ant ; discrete dynamical systems ; asynchronous updating ; robustness analysis

1 Introduction

This article is devoted to the study of a discrete-time system composed of parallel concurrent Turing machines that operate on a two-dimensional grid. The machines we consider are known as *turmites* [1] or Langton's ants [2]; they follow simple local rules, but their global behaviour is generally complex, even when only one agent is used. The complexity of the system has been well studied with a single turmite, either with the original rule [3, 4, 5] or with generalisations [6, 7]. However, systems with multiple turmites have been much less explored so far [8, 9]. One reason why the multi-turmite system has been scarcely studied is that introducing multiple turmites also produces ambiguities with regard to how to update agents and how to solve their potential conflicts. In some cases, ambiguities may even render experiments difficult to reproduce.

To tackle these difficulties, we follow the proposition of Chevrier and Fatès who introduced a method for describing multi-agent systems as discrete dynamical systems [10]. Discrete dynamical systems provide a formalism that is widely used to study complex systems such as cellular automata, neural networks, Boolean networks, etc. By contrast, there are other fields where the dynamical systems approach is much less used, for instance, the field of multi-agent systems. It was demonstrated that using the discrete dynamical systems approach enables to derive a panel of descriptions starting from a simple rule at the agent level [10]. Each description is obtained with a *simulation scheme*, that is, a particular way of updating components and a particular method for solving the potential conflicts that would appear during this updating. As a result, even when using the same model and when starting from the same initial condition, the use of different simulation schemes may produce several qualitative behaviours.

Research has been so far limited to considering different ways of dealing with the spatial conflicts that appear when multiple agents need to share the same location. This paper aims at extending this study by considering the case where the agents are updated asynchronously. Is synchronous update a necessary condition for observing interesting phenomena with turmites? In the case of cellular automata, studies have shown that asynchronous update leads to the observation of a wide range of surprising phenomena (see e.g., [11, 12, 13]). Our purpose is to present a similar study for the multi-turmite system. Can we observe new phenomena and estimate their robustness?

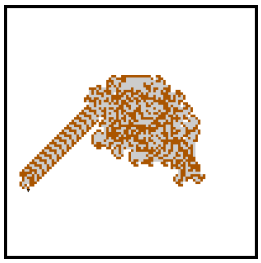


Figure 1: One Turmite drawing a Path

The outline of the paper is rather straightforward. Section 2 is devoted to the definition of the multi-turmite system and the different simulation schemes we study. In Sec. 3, we present three emergent phenomena and we study their robustness to asynchrony with a macroscopic and microscopic analysis. We then give a few perspectives opened by these observations.

2 Foundations

The system we consider uses the original turmite description as introduced by Langton [2] in the context of "artificial life" or Bunimovich and Troubetzkoy [14] in the context of particle systems. It consists of a two-dimensional square grid and a set of turmites. Each cell of the grid has one state: 0 or 1. A turmite is a special Turing machine which moves from one cell to a neighbouring cell in one of the four directions: North, East, South and West. Two symmetric rules define the evolution of the machine: If the turmite is on a cell in state 0 (respectively 1), the cell state flips to 1 (resp. to 0), the turmite turns left (resp. right) by 90° and advances to the next cell. The behaviour of a single turmite is surprising. It was observed that, starting from an initial grid with all cells in state 0, the turmite follows a non-structured trajectory on the grid for approximately 10000 steps and then suddenly enters into a periodic behaviour which leads to the formation of a regular translating structure called a *path* (or a "highway"; see Fig. 1).

Are similar or new behaviours observed when multiple turmites are simultaneously sharing the grid? Before we answer this question, let us first define formally our multi-turmite system. As we will see in the following, the operation is not as straightforward as it may first seem. Our presentation of the dynamical system follows the paper by Chevrier and Fatès [10], although we compact here all the intermediary steps of the method for the sake of conciseness.

We denote by $\mathcal{L} = \mathbb{Z}^2$ the grid. Each cell $c \in \mathcal{L}$ has a state in $\mathcal{Q} = \{0, 1\}$. The overall grid state is denoted by $\mathbf{S} \in \mathcal{Q}^{\mathcal{L}}$. Let N be the turmites number, it is supposed fixed. We denote by $\mathrm{T} = \{1, .., N\}$, the set of turmites. Each turmite i has a position $P_i \in \mathcal{L}$ and an orientation $O_i \in \{0, 1, 2, 3\}$ associated to the directions North, East, South, West, respectively.

We denote by $\mathbf{P} = (P_1, \ldots, P_N) \in \mathcal{L}^N$ and by $\mathbf{O} = (O_1, \ldots, O_N) \in \mathcal{D}^N$, the N-uple of turmites positions and orientations, respectively. The state of a system is a *configuration*; it is represented by a triplet $\sigma = (\mathbf{S}, \mathbf{P}, \mathbf{O}) \in \Sigma = \mathcal{Q}^{\mathcal{L}} \times \mathcal{L}^N \times \mathcal{D}^N$. Using this notations, we describe our multi-turmite system as a discrete dynamical system on Σ, that is, advancing by one time step corresponds to applying Γ, the *global transition function*: $\Gamma : \Sigma \to \Sigma$. Now, let us consider the following problem: How can we describe formally Γ given the informal description of the turmite behaviour (see above) and given that:

(a) we want to update the turmites synchronously or sequentially;

(b) we want to examine different ways of solving the conflicts that appear when multiple turmites simultaneously want to move on the same cell?

Our proposition consists of defining Γ with two auxiliary functions. The first function is the *updating method* Δ, the second function is the *conflict resolution policy* ξ. In this paper, we study three updating methods and three conflict resolution policies. Let us now present these functions formally.

The updating method is a function $\Delta : \mathbb{N} \to \mathcal{P}(\mathrm{T})$; it selects at each time step a (sub)set of turmites to update. We use three different updating methods Δ:

• *synchronous* update: all turmites are updated at each time step. We thus have:

$$\forall t \in \mathbb{N},\ \Delta_{\mathrm{s}}(t) = \mathrm{T}$$

• *cyclic* update: the turmites are updated sequentially in a fixed order following a cycle ; we use

$$\forall t \in \mathbb{N},\ \Delta_{\mathrm{c}}(t) = t \bmod N + 1$$

• *random* update: the turmites are also updated sequentially but the order of the updating within each cycle varies randomly. To define this updating method, we use

$$\forall t \in \mathbb{N},\ \Delta_{\mathrm{r}}(t) = \pi_{\lfloor t/N \rfloor}(t \bmod N + 1)$$

where $(\pi_k)_{k\in\mathbb{N}}$ is a series of independent random variables that draw an element uniformly in Π_{T}, the set of all permutations from T to T.

Now that we have defined Δ, let us now define Γ by specifying, independently, $\mathbf{S}$ on the one hand and, $\mathbf{P}$ and $\mathbf{O}$ on the other hand. At each time step, the grid state evolves following:

$$\forall c \in \mathcal{L}, S_c(t+1) = \begin{cases} 1 - S_c(t) & \text{if } c \in \{P_i, i \in \Delta(t)\} \\ S_c(t) & \text{otherwise} \end{cases}$$

This rule means that the state of a cell is changed if the cell contains at least one turmite. Other policies are possible, for instance considering the so-called "annihilation policy" where simultaneous flips are combined by pairs [10].

Let us now describe how to update the positions and orientations of the turmites. Defining $(\mathbf{P}, \mathbf{O})(t)$ requires to specify how turmites interact. Note that if a turmite is not updated, its position and orientation are unchanged: $\forall i \notin \Delta(t), (P_i, O_i)(t+1) = (P_i, O_i)(t)$. For an updated turmite $i \in \Delta(t)$, the way to calculate $(P_i, O_i)(t+1)$ depends on the conflict resolution policy ξ. We now define three different conflict policies ξ.

Allow policy ***(ξ_{AL})***. This policy allows turmites to move and rotate freely without taking into account conflicts. For a turmite i, starting from its orientation and position at time t, we denote by $\tilde{O}_i(t)$ and $\tilde{P}_i(t)$ the new orientation and position of this turmite if it were alone on the grid. This gives:

$$\tilde{O}_i(t) = \begin{cases} (O_i(t) + 1) \bmod 4 & \text{if } S_{P_i(t)}(t) = 0 \\ (O_i(t) - 1) \bmod 4 & \text{if } S_{P_i(t)}(t) = 1 \end{cases}$$

and

$$\forall i \in \mathrm{T}, \tilde{P}_i(t) = P_i(t) + \mathrm{dP}(\,\tilde{O}_i(t)\,)$$

where dP(.) is the function defined by $\mathrm{dP}(0) = (0,1)$, $\mathrm{dP}(1) = (1,0)$, $\mathrm{dP}(2) = (0,-1)$, $\mathrm{dP}(3) = (-1,0)$ (North, East, South and West translations).

The Allow policy then writes:

$$\xi_{\mathrm{AL}} : \forall i \in \Delta(t),\ (P_i, O_i)(t+1) = (\tilde{P}_i, \tilde{O}_i)(t)$$

Exclude policy ***(ξ_{EX})***. When a conflict occurs, the ξ_{EX} prohibit turmites move and rotation.
Those conflicts occur in two cases:

- type A: a turmite asks to move to an occupied cell.
- type B: two or more turmites ask to move to the same target cell.

To express these conflicts formally, we use a function n that counts the number of turmites present in a cell $c \in \mathcal{L}$ given a set of turmites positions $\mathbf{P}$:

$$\begin{array}{rccl} n: & \mathcal{L}^N \times \mathcal{L} & \to & \mathbb{N} \\ & n[\mathbf{P}, c] & \to & \mathrm{card}\{i \in \mathrm{T}, P_i = c\} \end{array}$$

Figure 2 shows an example of *type A* and *type B* conflict. The positions and orientations of turmite 1 and turmite 2 represent a *type B* conflict. Turmite 1 and 2 ask to move to the same cell c_1. The positions and orientations of turmite 3 and turmite 4 represent a *type A* conflict. Turmite 3 asks to move to the cell c_2 which contains turmite 4.

A movement of turmites that avoids type A conflicts is given by the condition: $n[\mathbf{P}, \tilde{P}_i] = 0$; a movement of turmites that avoids type B conflicts is given by the condition: $n[\tilde{\mathbf{P}}, \tilde{P}_i] = 1$. The Exclude policy then writes:

$$\xi_{\mathrm{EX}} : (P_i, O_i)(t+1) = \begin{cases} (\tilde{P}_i, \tilde{O}_i)(t) & \text{if } n[\mathbf{P}, \tilde{P}_i] = 0 \text{ and } n[\tilde{\mathbf{P}}, \tilde{P}_i] = 1 \\ (P_i, O_i)(t) & \text{otherwise} \end{cases}$$

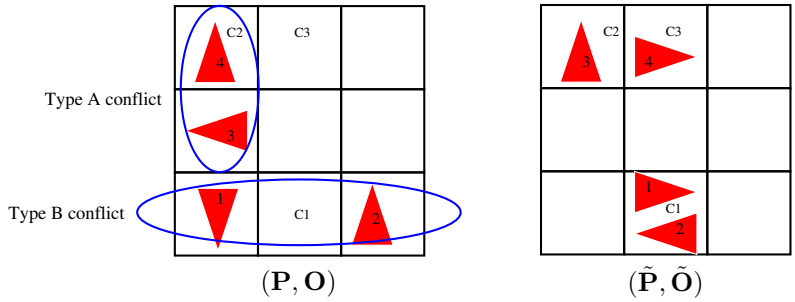


Figure 2: From the current position and orientation $(\mathbf{P}, \mathbf{O})$ of the turmites to their expected next position and orientation $(\tilde{\mathbf{P}}, \tilde{\mathbf{O}})$. The situation generates a type A conflict in $c_2 = P_4 = \tilde{P}_3$ and a type B conflict in $c_1 = \tilde{P}_1 = \tilde{P}_2$ and no conflict in $c_3 = \tilde{P}_4$. We have $n[\tilde{\mathbf{P}}, c_1] = 2$, $n[\mathbf{P}, c_2] = 1$, $n[\mathbf{P}, c_3] = 0$ and $n[\tilde{\mathbf{P}}, c_3] = 1$.

Turn and See policy ***(ξ_{TS})***. This policy is somewhat an intermediary policy between the Allow and Exclude policies. The turmites involved in a conflict do not move, but they are allowed to turn. Using the previous notations, this writes:

$$\xi_{\text{TS}} : \begin{cases} P_i(t+1) = \begin{cases} \tilde{P}(t) & \text{if } n[\mathbf{P}, \tilde{P}_i] = 0 \text{ and } n[\tilde{\mathbf{P}}, \tilde{P}_i] = 1 \\ P_i(t) & \text{otherwise} \end{cases} \\ \\ O_i(t+1) = \tilde{O}_i(t) \end{cases}$$

In short, we have 3 updating methods and 3 conflict resolution methods, which define 9 possible combinations and thus 9 dynamical systems Γ that we call *submodels*, since they derive from one single general simulation model. We denote a submodel by $\Gamma_{\Delta,\xi}$ where Δ is the updating method and ξ is the conflict resolution policy. Finally, we define a system *orbit* (or trajectory) as the sequence of configurations $(\sigma(t))_{t\in\mathbb{N}} = \text{ORB}(\Delta, \xi, \sigma)$ obtained with: $\sigma(0) = \sigma$ and $\forall t \in \mathbb{N}$, $\sigma(t+1) = \Gamma_{\Delta,\xi}(\sigma(t))$.

Having multiple submodels of simulation schemes does not prove by itself the importance of formalising simulation schemes. This importance can only be estimated through its effects, i.e., if it qualitatively modifies the orbits. We thus ask what are the interesting emergent phenomena that can be observed with different submodels.

3 Orbits and Robustness to Asynchrony

Now that we transformed our system from an informal individual-based description to a dynamical systems description, let us observe the perspectives opened by this change of viewpoint. As a first step, we focus our attention on the asymptotic evolution of the orbits. This problem is too wide to be tackled for all discrete dynamical systems and it is necessary to define classes of behaviour that are specific to our system and to our interests. For instance, the question of classification of cellular automata has been challenging authors for more two decades (see e.g. [15]). In this section, we propose a typology of orbits to classify the behaviours of our system. We then study three emergent phenomena, that is, interesting behaviours which can not be predicted simply by looking at the local rules that define the system. We analyse how these phenomena are affected by changes in the simulation schemes (updating method and conflict resolution policies).

3.1 Typology of Orbits

Recall that orbits are denoted by $(\sigma(t))_{t\in\mathbb{N}} = \text{ORB}(\Delta, \xi, \sigma)$, where $\sigma(t) = (\mathbf{S}, \mathbf{P}, \mathbf{O})(t)$. We now present a set of definitions aimed at separating the different types of orbits that can be obtained with our system. The definitions are analogous, but not identical, to the ones proposed by Gajardo and Goles in [7].

Definition 1 (Traces) *For an orbit $(\sigma(t))_{t\in\mathbb{N}}$, the* trace ζ_i *of a turmite i is the set of cells visited by this turmite: $\zeta_i = \{P_i(t) \in \mathcal{L}, t \in \mathbb{N}\}$. The trace of an orbit is the union of all traces of the turmites.*

Definition 2 (Support) *The* support η *of an orbit* $(\sigma(t))_{t\in\mathbb{N}}$ *is the set of cells visited infinitely often by the turmites:*

$$\eta = \cup_{i\in \mathrm{T}}\{c \in \mathcal{L}, |\{t, P_i(t) = c\}| = \infty\}$$

Definition 3 (Cycle) *An orbit* $(\sigma(t))_{t\in\mathbb{N}}$ *is a* cycle *if*

$$\exists\, t', p \in \mathbb{N},\, \forall t \geq t',\, \sigma(t + p) = \sigma(t)$$

The smallest t' and p for which the property is verified are called the *transient time* and the *period* of a cycle.

Definition 4 (Translation) *For an orbit* $(\sigma(t))_{t\in\mathbb{N}}$*, we say that a turmite* $i \in T$ *is in an infinite* translation *if*

$$\exists \Delta P \in \mathcal{L}, \exists p \in \mathbb{N}, \exists t_0 \in \mathbb{N}, \forall t \geq t_0,\, P_i(t + p) = P_i(t) + \Delta P \text{ and } O_i(t + p) = O_i(t)$$

We say that an orbit is in an infinite translation if all its turmites are in infinite translation.

Definition 5 (Deadlock) *An orbit* $(\sigma(t))_{t\in\mathbb{N}}$ *is a* deadlock *if*

$$\exists t', \forall t \geq t', \forall i \in \mathrm{T},\, P_i(t) = P_i(t')$$

The smallest t' for which the property is verified is called the *deadlock time*.

Using, the definitions above we now introduce a typology to classify the behaviour of our system:

- We say that an orbit is *bounded* if its trace is finite, let $\mathcal{B}$ be the set of bounded orbits.
- $\mathcal{S}$ is the set of orbits with a finite support. By convention, an orbit with an empty support is in $\mathcal{S}$.
- $\mathcal{C}, \mathcal{T}, \mathcal{D}$ are the sets of orbits which are cycles, translations and deadlocks, respectively.

We now examine the relationships between these sets. First, we can note that a bounded orbit has a finite support but the converse is not necessarily true. We can for instance imagine a case where the turmites return infinitely often to the central cell but explore a lager set of cells after each of their return (this could be achieved with a proper setting of the state of the cells in the turmites trace). We thus have $\mathcal{B} \subset \mathcal{S}$.

Similarly, we should note that $\mathcal{C} \subset \mathcal{B}$, that is, a cycle is a bounded orbit but the converse is not necessarily true. For instance, the turmites of a bounded orbit obtained with a random updating method (Δ_r) might evolve randomly on a limited part of the grid. However, if the updating method is synchronous (Δ_s), or follows a cycle (Δ_c), then a bounded orbit is a cycle.

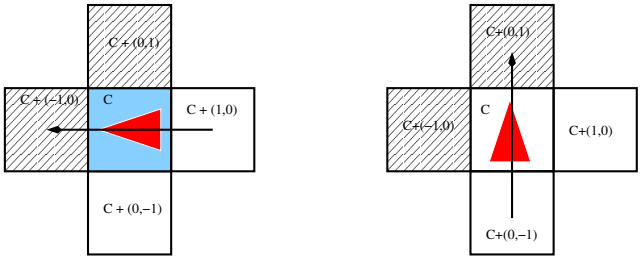
To prove this assertion, let us first consider the case of synchronous updating. Let us assume that the trace of the orbit, denoted by η, is finite and has size $s = |\eta|$. We define $E = \{\sigma(t), t \in \mathbb{N}\}$ as the set of configurations of the orbit. As η is finite, E is also finite and we have $|E| \leq 2^s \cdot (4s)^N$. Indeed, in each configuration of E, the cells out of η have the same state, the cells in η can have two states and each turmite can be in one of the 4 orientations and on one cell in η. We can thus deduce that there exist two time steps $t_1 < t_2$ such that $\sigma(t_1) = \sigma(t_2)$. As the evolution of the system is deterministic, we have $\forall t \in \mathbb{N}$, $\sigma(t_1 + t) = \sigma(t_2 + t)$, from which we can derive $\forall t \in \mathbb{N}$, $\sigma(t_1 + (t_2 - t_1) + t) = \sigma(t_1 + t)$, which implies that the orbit is a cycle (whose period divides $t_2 - t_1$ and whose transient time is smaller or equal to t_1). This proof also holds for the cyclic update case, provided that we consider "super-configurations" which include the time t mod N and define the set E as $\{(\sigma(t), t \bmod N), t \in \mathbb{N}\}$.

It straightforward to see that $\mathcal{T} \subset \mathcal{S}$ and that $\mathcal{D} \subset \mathcal{C}$. We thus obtain the following inclusions of the sets of orbits : $\mathcal{D} \subset \mathcal{C} \subset \mathcal{B} \subset \mathcal{S}$ and $\mathcal{T} \subset \mathcal{S}$. Using this typology, we say that the phenomenon produced by an initial condition is *robust* if the orbits which are obtained with the different updating methods are in the same set of orbits.

3.2 Cycles and Clocks

The first phenomenon that caught the attention of the researcher was that a mono-turmite system starting from an all-0 grid leads to the construction of a *path* structure, i.e., a translating orbit in which the trace of the turmite has a regular pattern. One may thus wonder whether a single turmite might construct other regular structures. For example, is it possible to observe a mono-turmite orbit that is a cycle?

The answer to this question is negative [14, 8]. To prove this assertion, let us assume that the orbit is bounded and show that it leads to a contradiction. As seen above, the orbit has a finite support; let B be its



First visit of corner cell c Second visit of corner cell c

Figure 3: Illustration of the two cases which cause a contradiction in the proof that a single turmite can not establish a cycle. Hatched cells do not belong to the support of the cycle. White and blue colours represent 0 and 1 cells, respectively.

support[1]. As the turmite turns by 90° at each time step, B can not be reduced to a line; there are thus at least four corners in B. Without loss of generality, let us consider a "top-left corner cell", that is, a cell $c \in B$ that verifies: $c + (0,1) \notin B$, $c + (-1,0) \notin B$, $c + (1,0) \in B$ and $c + (0,-1) \in B$. By the definition of B, there exist two instants t_1 and t_2 such that $P(t_1) = c$ and $P(t_2) = c$ and $S_c(t_1) = 1 - S_c(t_2)$ (see Fig. 3).

We prove the existence of a contradiction with parity arguments. The *position parity* is the function $\pi_{\mathrm{P}}(t) = (P_x(t) + P_y(t)) \bmod 2$ and the *orientation parity* is $\pi_{\mathrm{O}}(t) = O(t) \bmod 2$. From the definition of the local rule, we can deduce: $\pi_{\mathrm{P}}(t+1) = 1 - \pi_{\mathrm{P}}(t)$, $\pi_{\mathrm{O}}(t+1) = 1 - \pi_{\mathrm{O}}(t)$ and a global parity conservation property:

$$\forall t \in \mathbb{N},\ \pi_{\mathrm{P}}(t) + \pi_{\mathrm{O}}(t) = \pi_{\mathrm{P}}(0) + \pi_{\mathrm{O}}(0) \bmod 2$$

We have $P(t_1) = P(t_2)$, and thus $\pi_{\mathrm{P}}(t_1) = \pi_{\mathrm{P}}(t_2)$. The global parity conservation gives: $\pi_{\mathrm{O}}(t_1) = \pi_{\mathrm{O}}(t_2)$. On the other hand, as seen above $S_c(t_1) = 1 - S_c(t_2)$; we thus have $\pi_{\mathrm{O}}(t_1) = 1 - \pi_{\mathrm{O}}(t_2)$, which contradicts the previous result. This proof was already established by other authors [14, 8] but it is here easy to formulate mathematically with the formalism introduced above.

Interestingly, for the multi-turmite system and for particular initial conditions, it is possible to observe cycles. First observations of cycles were reported in the experimental study by Chevrier and Fatès [10]. However, it is in general difficult to predict the length and the support of a cycle as a function of the initial condition. We now present a particular set of initial conditions for which this prediction is possible, forming a phenomenon that we call a *clock*.

Observation 1 *For an even number of turmites N, for an initial condition $\sigma = \{\mathbf{S} = 0, \forall i \in \mathrm{T},\ O_i = 0, P_i = (i-1, 0)\}$, the orbit $\mathrm{ORB}(\Delta_{\mathrm{s}}, \xi_{\mathrm{AL}}, \sigma)$ is a cycle.*

We experimentally determined that the length of the cycle $l(N)$ varies as $l(N) = 16N - 4$ and that the transient time is 0. Moreover, the support of the cycle can be enclosed in a rectangular zone of $3N \times 2N$ cells.

Figure 4 shows two cycles with 4 and 12 turmites. The configurations at time $t = 60$ and $t = 188$, respectively, shows the end of the cycle, i.e., when the configuration is identical to the the initial configuration.

Robustness For the set of initial conditions described above, the clock phenomenon is only observed with the $\Gamma_{\Delta_{\mathrm{s}}, \xi_{\mathrm{AL}}}$ submodel, i.e., with the synchronous update and the Allow policy (see Tab. 1). For all other systems considered, no regularity of behaviour was observed, at least for the first few hundred steps of evolution. Moreover, the divergence between the evolution of the different submodels is observed only after a few dozens of steps. How can we explain this sensitivity?

Microscopic analysis To explain the sensitivity of the phenomenon to the asynchronous updating and to the change of conflict resolution policy, we first report that the orbit contains a particular type of conflict, the *break* conflict, characterised by the pattern presented on Fig. 5 and by the symmetrical patterns. We note that:

(a) if the ξ_{TS} or ξ_{EX} policies are used, a divergence appears after only one step since the movements of the two turmites are prohibited in type B conflicts.

[1]Remark that for one turmite, all updating methods are equivalent to synchronous updating, the orbit of the system is thus a cycle (see above).

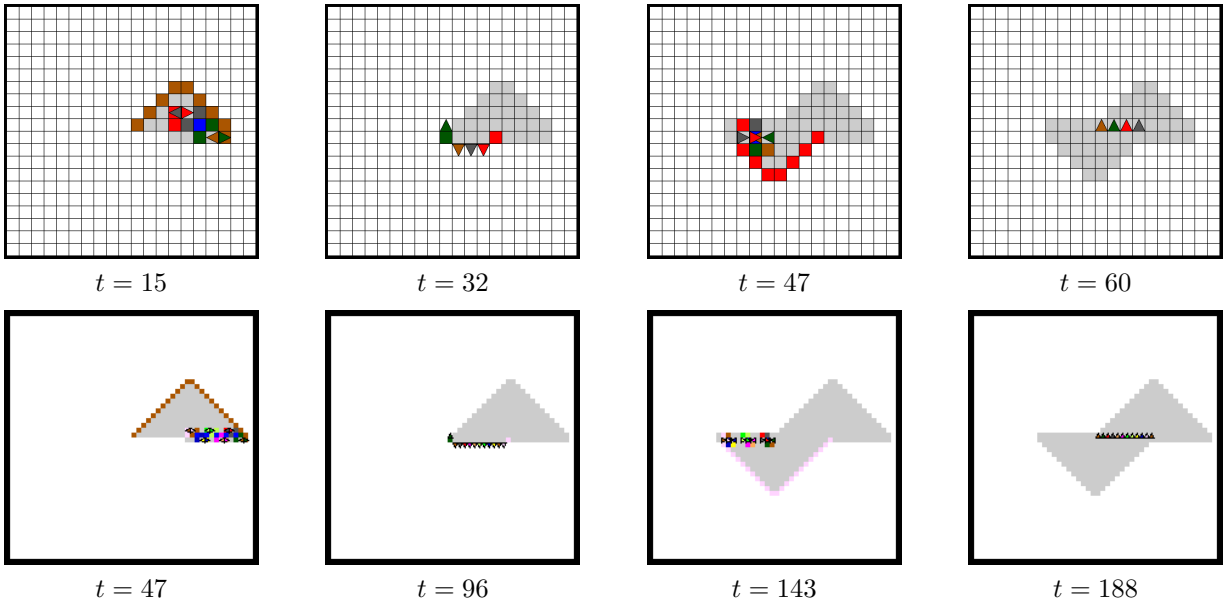


Figure 4: Clock cycle with: (top) 4 Turmites, (bottom) 12 Turmites. Each turmite has its own colour and gives this colour to the cells in state 1 when it flips a cell's state from 0 to 1. Cells in state 0 are initially in white and are shown in light grey if they have been flipped at least once.

(b) In the case of ξ_{AL} policy, the divergence with asynchrony appears after two update cycles (here, an update of the two turmites). After the first update cycle, the evolution of the two systems is similar. However, at the next update cycle, the two turmites leave the cell with opposite directions in the synchronous mode while they leave it with identical directions in the asynchronous mode.

These results are reported on Tab. 2.

3.3 Gliders

Gliders are rare phenomena in the multi-turmite system; it was necessary to test for thousands of different configurations for observing them. They are "purely translating" patterns, where the turmites go in a straight direction and leave a traces with cells in state 0.

Definition 6 *An orbit* $\mathrm{ORB}(\Delta, \xi, \sigma)$ *is a* glider *if it is in an infinite translation and if*

$$\exists \Delta t, \forall t' > \Delta t, \forall i \in \mathrm{T},\ S_{P_i(t)}(t + t') = 0$$

These gliders are phenomena which are analogous to the gliders observed in the Game of life cellular automaton [16]. Only two gliders have been detected so far. The first glider is the *F-glider* [10] and the second one is the *B-glider*, described in the property below:

Observation 2 *For* $N = 4$, *for* $\sigma = \{\mathbf{S} = 0, \forall i \in \mathrm{T},\ O_i = 0, P_0 = (0,0), P_1 = (1,0), P_2 = (1,1), P_3 = (0,1)\}$ *and for* $\Delta \in \{\Delta_{\mathrm{s}}, \Delta_{\mathrm{c}}, \Delta_{\mathrm{r}}\}$, $\mathrm{ORB}(\Delta, \xi_{\mathrm{AL}}, \sigma)$ *is a glider.*

Figure 6 shows four steps in the evolution of the B-Glider; this glider translates with a distance of 2 cells (horizontally or vertically) every 12 steps.

Robustness Unlike the F-glider, the B-glider is robust to changes in the updating method as well as changes in the conflict resolution policy (see Table 1).

Microscopic analysis To understand the robustness of the B-glider, we analysed its evolution and found out that involves only one form of conflict. The steps where this conflict appear on Fig. 6 at times $t = 4$ and $t = 8$.

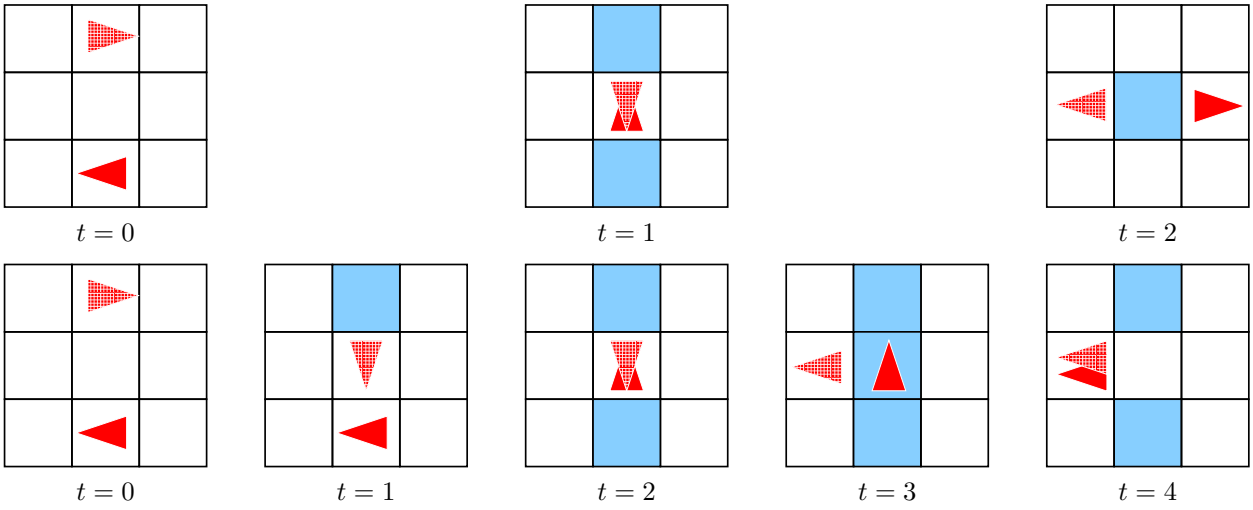


Figure 5: Microscopic analysis showing the non-robustness of the "break" conflict. Allow policy with: (top) synchronous, (bottom) asynchronous update. The hatched turmite is updated before the plain turmite.

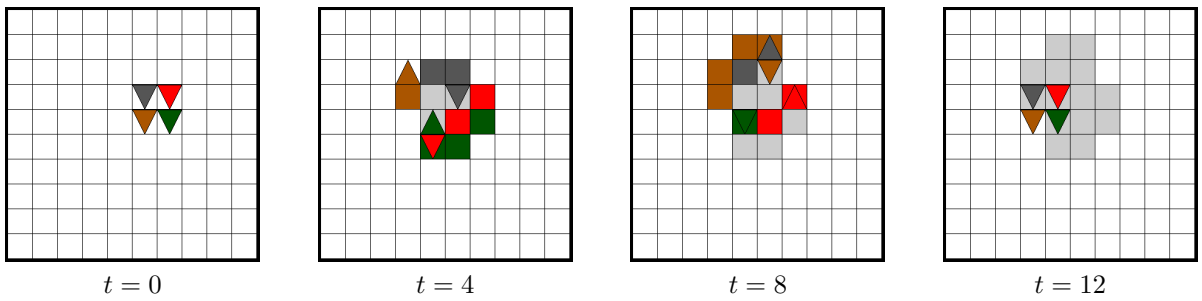


Figure 6: Four steps in the translation cycle of the B-glider.

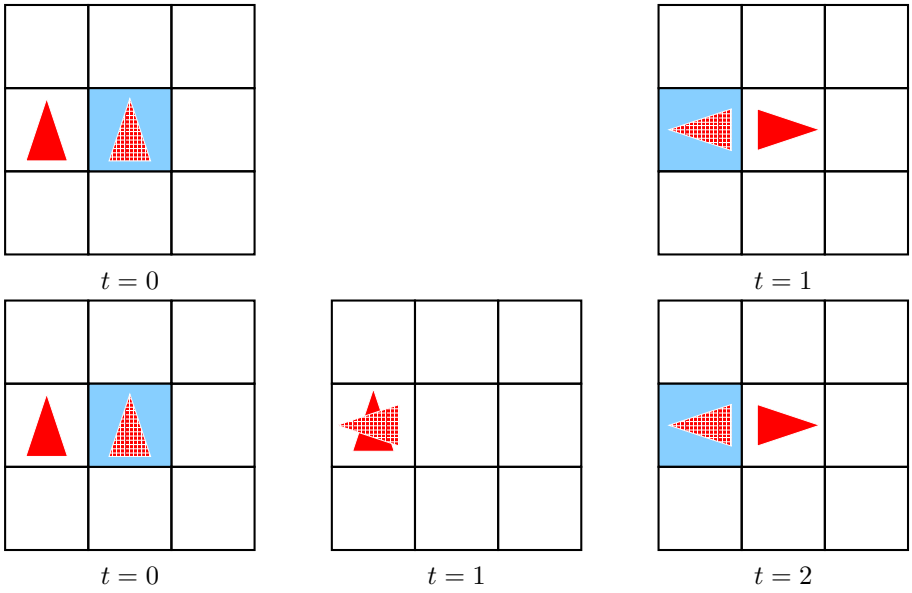


Figure 7: Microscopic analysis showing the robustness of the "reversion" conflict. Allow policy with: (top) synchronous update, (bottom) asynchronous update.

We call this particular form of type A conflict the *reversion conflict*. For two turmites a and b, it is characterised by the pattern: $O_a = O_b$, $P_b = P_a + (1, 0)$, $S_a = 0, S_b = 1$ and by the three other equivalent patterns obtained by 90° rotations (see Fig. 7).

As this is a conflict of type A, the ξ_{AL} submodel allows the turmites to exchange their positions. With the asynchronous update, the exchange happens in two steps but the result is the same, whatever the order of updating of the turmites (see Fig. 7). This similar evolution explains the robustness of the glider to the asynchronous update. Clearly, the ξ_{TS} and ξ_{EX} submodels have a different behaviour since type A conflicts imply a divergence in the evolution of the turmites (their positions are not modified, but the state of their cells is). The results of the microscopic analysis are reported on Tab. 2.

Remarkably, this behaviour generally leads to the production of cyclic or translating orbits. When this conflict appears, its effect results in the inversion of the roles of the two turmites. After the conflict has occurred, we generally observe that each turmite erases the trace left by the other turmite. This phenomenon has already been observed by other authors (e.g., Chopard and Droz [8], Chevrier and Fatès [10]) but was only partly explained.

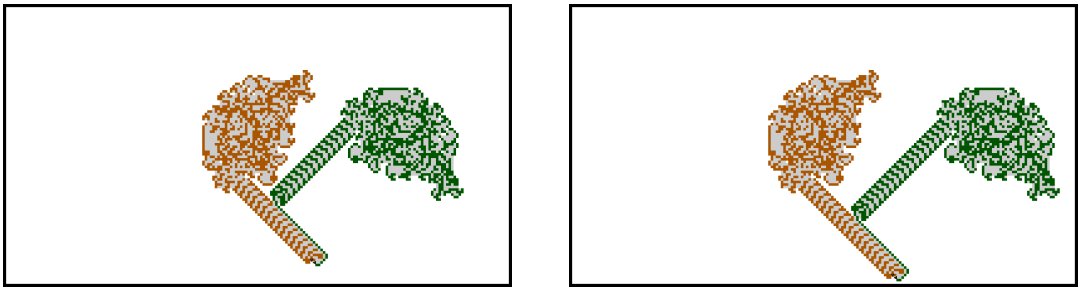
3.4 Deadlock

We now present our third class of initial conditions, which produces a phenomenon that we call a *deadlock*. It was originally observed for the synchronous updating and for the Turn and see policy.

Observation 3 *For $N = 2$, for $k \in 4\mathbb{N}^*$ and $k \geq 52$ and $\sigma_k = \{\mathbf{S} = 0, O_1 = 3, O_2 = 0, P_0 = (0, 0), P_1 = (k, 0)\}$, the orbit $\mathrm{ORB}(\Delta_{\mathrm{s}}, \xi_{\mathrm{TS}}, \sigma_k)$ is a deadlock.*

The initial condition is made of two turmites initially heading West and North and separated by k cells. Figure 8 shows two deadlocks obtained with two distances k. Note that although deadlocks are also observed for $k < 52$, there is no regularity in the way the phenomenon happens. By contrast, for $k > 52$, we can guarantee that the two turmites always interact in the same way. The minimal "security" distance $k = 52$ is given to ensure that each turmite does generate its own path and is not "perturbed" by an overlap of its trace and the trace of the other turmite. If this condition is met, turmite 2 always crosses the path of turmite 1 the same way: it follows the path and surrounds it and then catches up with turmite 1. This produces a type B conflict, which always results in a deadlock.

Increasing the value of k with 4 cells ensures that the two turmites follow the same sequence of behaviour before they meet. This regularity is due to the spatial periodicity of the path construction, which is invariant by a translation of $(2, -2)$. We also observe that an increase of k by 4, increases the deadlock time by



$k = 52$ $k = 64$

Figure 8: Two deadlock situations obtained with 52 and 64 initial interspace between the two turmites (see text for a precise description of the initial condition).

Table 1: Robustness of the three phenomena observed with 9 submodels.

submodel	ξ_{AL}			ξ_{EX}			ξ_{TS}		
	Δ_{s}	Δ_{c}	Δ_{r}	Δ_{s}	Δ_{c}	Δ_{r}	Δ_{s}	Δ_{c}	Δ_{r}
clock	yes	no	no	no	no	no	no	no	no
B-glider	yes	yes	yes	no	no	no	no	no	no
deadlock	no	no	no	no	no	no	yes	no	no

Table 2: Robustness of three conflict forms observed with 9 submodels.

submodel	ξ_{AL}			ξ_{EX}			ξ_{TS}		
	Δ_{s}	Δ_{c}	Δ_{r}	Δ_{s}	Δ_{c}	Δ_{r}	Δ_{s}	Δ_{c}	Δ_{r}
break	yes	no	no	no	no	no	no	no	no
reversion	yes	yes	yes	no	no	no	no	no	no
dual lock	no	no	no	no	no	no	yes	no	no

104. These experimental values confirm that the invariance of the sequence of behaviour as they precisely correspond to the values obtained by Boon with analytical arguments [4]. We can thus predict the deadlock time t_{deadlock} as a function of k with: $t_{\mathrm{deadlock}}(k) = t_{\mathrm{deadlock}}(52) + \frac{(k-52)}{4} \times 104$, using the experimental value $t_{\mathrm{deadlock}}(52) = 11902$.

Robustness Deadlocks were observed only with the $\Gamma_{\Delta_{\mathrm{s}},\xi_{\mathrm{TS}}}$ synchronous Turn and see submodel (see Tab. 1). The sensitivity of this phenomenon to asynchrony comes from the fact that this type of orbit contains only one form of conflict, the *dual lock* conflict.

Microscopic analysis For two turmites a and b, the dual lock pattern is characterised by a type B conflict with two turmites with perpendicular orientations. This can be seen with the pattern: $P_b = P_a + (1, -1)$, $O_a = 1$, $O_b = 0$, $S_a = 0$, $S_b = 1$ and by the symmetrical patterns (see Fig. 9). In this case, a type B conflict occurs on the cell $c = \tilde{P}_a = \tilde{P}_b$. With the Turn and see policy the new orientations and positions are: $O'_a = 2$, $O'_b = 3$ and $P'_a = P_a$, $P'_b = P_b$ (the positions of the turmites are unchanged). Then, the same type of conflict appears on the cell $c' = \tilde{P}'_a = \tilde{P}'_b$ and the turmites are again in a type B conflict that results in a deadlock. With an asynchronous updating, turmite 1 moves before turmite 2, which "frees" the deadlock (see Fig. 9).

Clearly no deadlock can occur with an ξ_{AL} submodel ; with the ξ_{EX} submodels, turmites keep the same orientation and position but the state of their cell changes. This brings them to move in the opposite direction and solves the conflict. The dual lock conflict thus has a diverging outcome with ξ_{TS} and ξ_{EX} submodels (results reported on Tab. 2). In general, we observed that orbits that involve type B conflicts are not robust to the introduction of asynchronous updating.

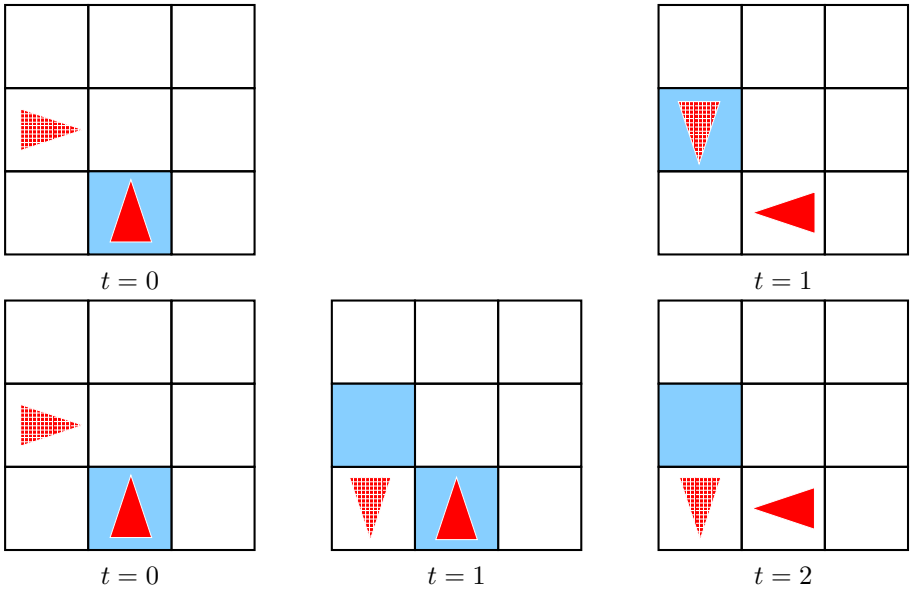


Figure 9: Microscopic analysis of the "dual lock" conflict. Turn and see policy with: (top) synchronous update, (bottom) asynchronous update.

4 Discussion

We presented clocks, gliders and deadlocks as three emergent phenomena in the multi-turmite system. Their robustness was tested with nine different simulation scheme (see Tab. 1 for a synthesis). These simulation schemes were defined with a dynamical systems approach, as a combination of the updating method and the conflict resolution policy. This allowed us to have a non-ambiguous description of the interactions in the system, a criterion which needs to be respected for the sake of the reproducibility of the experiments. Our formalism also allowed to define various types of orbits and thus to give a rigorous — although partial — definition of the robustness of those phenomena to asynchrony. Interestingly, we found a relevant correlation between the robustness of the orbits and the conflicts that occurred during the turmites movements. This correlation was explained with a microscopic analysis of the conflicts (see Tab. 2 for a synthesis).

If the classical Langton's ant (mono-turmite) system is still not fully understood, it appears that the multi-turmite system opens a wide realm which remains yet to be explored. Many other puzzling phenomena deserve to be studied. An interesting example is the so-called *ever growing square* (or diamond) where the turmites collaborate to produce a pattern that progressively expands (see [2, 10]). A quick analysis of this phenomenon lead us to conclude that its emergence is more difficult to predict than the three phenomena we presented.

The strong relationship that we noticed between the initial condition, the conflicts forms and the type of asymptotic behaviour of the system also calls for further analysis. The robustness or sensitivity of the phenomena to the various simulation schemes could be here explained as a result of the robustness of some conflict patterns. A challenging problem is now to derive stronger relationships to predict the robustness of a phenomenon given the conflicts that it involves, for instance with a proper analysis of the initial condition (symmetries, conserved quantities, etc.)

Finally, coming back to the fact that the multi-turmite system has a some physical inspirations, we ask to which extent the system can be used as a computing device. Gajardo *et al.* already shown that a single turmite was capable of universal computations [5]. The multi-turmite system may provide an alternative construction for bits of information, memories and logical gates. For instance, the two gliders could be used as in Conway's construction of a Turing machine in the *Game of Life* [16]. The clock phenomenon we presented might well be used for synchronising the computations but there is still the need to examine if clocks and gliders might interact in a controlled way. In the same spirit, deadlocks could be used as a mechanism either for storing or blocking information.

References

[1] A. K. Dewdney, Two-dimensional Turing machines and tur-mites make tracks on a plane, computer recreations, Scientific American (1989) 180–182.

[2] C. G. Langton, Studying artificial life with cellular automata, Physica D 22 (1986) 120–149.

[3] D. Gale, J. Propp, S. Sutherland, S. Troubetzkoy, Further travels with my ant, Mathematical Entertainments column, Mathematical Intelligencer 17 (1995) 48–56.

[4] J. P. Boon, How fast does Langton's ant move?, Journal of Statistical Physics 102 (2001) 355–360.

[5] A. Gajardo, A. Moreira, E. Goles, Complexity of Langton's ant, Discrete Applied Mathematics 117 (1-3) (2002) 41 – 50.

[6] A. Gajardo, E. Goles, A. Moreira, Generalized Langton's ant: Dynamical behavior and complexity, in: A. Ferreira, H. Reichel (Eds.), Proceedings of STACS 2001, Vol. 2010 of Lecture Notes in Computer Science, Springer, 2001, pp. 259–270.

[7] A. Gajardo, E. Goles, Dynamics of a class of ants on a one-dimensional lattice, Theoretical Computer Science 322 (2) (2004) 267–283.

[8] B. Chopard, M. Droz, Cellular automata modeling of physical systems, Cambridge University Press, 1998, pp. 46–51.

[9] O. Beuret, M. Tomassini, Behaviour of multiple generalized Langton's ants, in: C. Langton, K. Shimohara (Eds.), Proceeding of the Artificial Life V Conference, MIT Press, Nava, Japan, 1998, pp. 45–50.

[10] V. Chevrier, N. Fatès, Multi-agent Systems as Discrete Dynamical Systems: Influences and Reactions as a Modelling Principle, Research report, short version to appear in the Proceedings of AAMAS'10 (Dec. 2008).
URL `http://hal.inria.fr/inria-00345954/en/`

[11] N. Fatès, M. Morvan, N. Schabanel, E. Thierry, Fully asynchronous behavior of double-quiescent elementary cellular automata, Theoretical Computer Science 362 (2006) 1–16.

[12] N. Fatès, L. Gerin, Examples of fast and slow convergence of 2D asynchronous cellular systems, Journal of Cellular Automata 4 (4) (2009) 323–337.

[13] D. Regnault, N. Schabanel, E. Thierry, Progresses in the analysis of stochastic 2D cellular automata: A study of asynchronous 2D minority, Theoretical Computer Science 410 (47-49) (2009) 4844–4855.

[14] L. A. Bunimovich, S. E. Troubetzkoy, Recurrence properties of lorentz lattice gas cellular automata, Journal of Statistical Physics 67 (1992) 289–302.

[15] G. Cattaneo, E. Formenti, L. Margara, Topological chaos and cellular automata, in: M. Delorme, J. Mazoyer (Eds.), Cellular Automata - A Parallel model, Vol. 460, Kluwer Academic Publishers, 1999, pp. 213–259.

[16] E. R. Berlekamp, J. H. Conway, R. K. Guy, Winning Ways for your Mathematical Plays, Vol. 2, Academic Press, ISBN 0-12-091152-3, 1982, chapter 25.