Robustness of multi-agent models: the example of collaboration between turmites with synchronous and asynchronous updating

Selma Belgacem, Nazim Fatès

Robustness of multi-agent models: the example of collaboration between turmites with synchronous and asynchronous updating

Selma Belgacem *and Nazim Fatès †
Inria Nancy – Grand Est & LORIA
615 rue du Jardin botanique, 54 600 Villers-lès-Nancy ; France

January 24, 2013

Abstract

The robusntess of multi-agent systems to simulation conditions is analysed through a precise example, invented by Langton to investigate the foundations of artificial life. This system is composed of simple and memoryless agents, the turmites, which obey simple discrete local rules. While the local rules that govern each agent are kept constant, the interaction between agents are modified through nine variations. Our method consists in varying the updating scheme (synchronous vs. asynchronous) and the local conflict resolution policy (strong or weak exclusion rules). We experimentally estimate the effect of these modifications on three collaborative phenomena. We also analyse how the dynamics at the microscopic scale reflects the robustness of the system at the macroscopic scale. Observations confirm that the definition of the agents' behaviour is not the only setting that matters in the emergence of collaborative phenomena in complex systems: the way the agents are updated is also a key choice.

FOREWORD

This is the preprint of the following paper :
Selma Belgacem and Nazim Fatès, Robustness of multi-agent models:
the example of collaboration between turmites with synchronous and asynchronous updating, *Complex systems*, Vol. 21 (3), 2012, pp. 165-182.
Please refer to the published paper for a corrected version.

*Now in PhD in LITIS laboratory, Rouen University, France. Electronic mail address: selma.belgacem@etu.univ-rouen.fr
†Corresp. author, nazim.fates@loria.fr

1 Introduction

Simple discrete models. In a pioneering paper on artificial life published in 1986, Langton stated that: "A common aggregate organisation in nature is that of *society*. The global behaviour of a society is an emergent phenomenon, arising out of all of the local interactions of its members. [...] We know that complex behaviour can emerge from the interaction of very simple parts. Colonies of social insects provide a good subject material to the study of artificial life because they so readily exhibit complex behaviour emerging from the interaction of very simple living parts." [18] The method followed by Langton was very close to Turing's work on morphogenesis [20]: Instead of trying to capture life's complexity by building more and more realistic models, which is a never-ending task, simplifying the model as much as possible can help us identify the mechanisms that are *sufficient* for a phenomenon to appear.

Langton proposed to consider a simple system composed of one or several agents that operate on a two-dimensional grid[1]. Each cell of the grid has one state: 0 or 1. The system is updated in discrete time and the agents, now known as *Langton's ants* or *turmites* [9], are memoryless and follow two simple symmetric laws:(a) If the turmite is on a cell in state 0, the cell state flips to 1, the turmite turns left and advances to the next cell. (b) If the turmite is on a cell in state 1, the cell state flips to 0, the turmite turns right and advances to the next cell.

Langton observed that the behaviour of single turmite was already a puzzling phenomenon. In the case where several agents where put together, interesting collaborative phenomena could emerge and lead to the construction of drastically different patterns that those observed for a single agent. This was interpreted as an emergence of collaboration between agents, a topic which is now widely considered as a key problem in many sciences. In biology for instance, it is still a challenge to understand how social insects may collaborate to construct their nests [17]. However, as Langton himself admits: "There are so many ways that these virtual ants can encounter one another that the transition rules have not yet been worked out for all of the possible encounters." Our goal in this paper is to complete the work of Langton and followers by broadening the way interactions between ants are defined.

We aim to discover not only novel collective phenomena, but we also wish to gain insight into how much the global behaviour of the system depends on these local interactions.

Turmites. The dynamics of turmites has been well studied when only a single turmite or particle is considered [16, 3, 13]. Starting from an initial grid with all cells in state 0, the turmite follows an irregular trajectory for approximately 10 000 steps and then suddenly enters into a periodic behaviour. This behaviour leads to the formation of a regular translating structure called a *path*

[1] Note that Bunimovich and Troubetzkoy have studied an equivalent system in the context of particle systems [4].

(or a "highway"; see Fig. 1). Different generalisations [15, 14] have also been proposed, but, surprisingly enough, systems with multiple turmites have been much less explored so far. The only results we are aware of are the studies by Chopard and Droz [8] and by Beuret and Tomassini [2].

Nine variations on one rule. One possible reason why the multi-turmite system has been scarcely studied is that introducing multiple turmites also produces ambiguities. Indeed, it is not clear from the local rule how to decide in which order (if any) to update agents and how to solve their potential conflicts when they share the same target cell. In some cases, these ambiguities may even render experiments difficult to reproduce.

To tackle these difficulties, a method was proposed by Chevrier and Fatès as a specification of Ferber and Muller's influence-reaction paradigm [12]. It consists in describing multi-agent systems as discrete dynamical systems [6]. 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 theme), the use of different simulation schemes (the variations) may produce several qualitative behaviours.

In the case of cellular automata, after the pioneering observations by Ingerson and Buvel [5], a number of studies have shown that asynchronous update leads to the observation of a wide range of surprising phenomena (see e.g., [11, 19, 10] for recent references). Our purpose is to present a similar study for a simple multi-agent system. We consider different ways of dealing with the spatial conflicts that appear when multiple agents need to share the same location and examine the difference produced with a synchronous and asynchronously update.

From artificial life to natural phenomena. Although the system proposed by Langton is simplistic, it may help us evaluate the effects of implicit choices in the simulation of more complex systems. For instance, if one needs to model biological systems such as viruses or bacteria, it may become possible to estimate separately how much of the observed behaviour is due to the internal dynamics of cells and how much is due to the interactions between cells. Moreover, in discrete systems, it is often difficult to decide whether to use a synchronous model (such as lattice-gas cellular automata) or an asynchronous model (such as interacting particle systems). In this work, we do not decide a priori but rather propose to test different simulation scenarios and compare them from a phenomenological point of view.

The outline of the paper is as follows. 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 approach. A microscopic analysis is then carried out in Sec. 4 to understand this robustness in more detail. Finally, we discuss

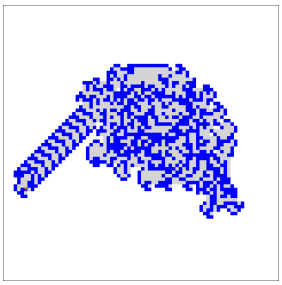


Figure 1: Evolution of a system with a single turmite starting from an empty environment: the turmite draws an infinite path. White and grey cells are non-visited and visited 0-cells, respectively; blue/dark cells are 1-cells. This convention is kept in the remainder of the paper.

the questions opened by these observations in Sec. 5.

2 Foundations

Before examining the outcome of simulations, 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 method of Chevrier and Fatès [6], although we skip here all the intermediary steps of the method for the sake of conciseness.

We denote by $\mathcal{L} = \mathbb{Z}^2$ the grid (or lattice). 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 number of turmites number, we denote by $\mathrm{T} = \{1, \ldots, N\}$ the set of turmites. Each turmite i has a position $P_i \in \mathcal{L}$ and an orientation $O_i \in \mathcal{D} = \{\mathtt{N}, \mathtt{E}, \mathtt{S}, \mathtt{W}\}$ 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-tuple of turmite 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 these 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 according to three temporal schemes;

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

Our proposition consists in defining Γ with two auxiliary functions. The first function is the *updating method* Δ, the second function is the *conflict resolution policy* ξ. We denote by $\Gamma_{\Delta,\xi}$ the global updating function obtained with an

updating method Δ and a conflict resolution policy ξ. The system's *orbit* (or trajectory) is the sequence of configurations $(\sigma(t))_{t\in\mathbb{N}} = \mathrm{ORB}(\Delta, \xi, \sigma)$ obtained with: $\sigma(0) = \sigma$ and $\forall t \in \mathbb{N}$, $\sigma(t+1) = \Gamma_{\Delta,\xi}(\sigma(t))$. In this paper, we study three updating methods and three conflict resolution policies. Let us now present these functions with both informal and formal definitions.

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

• *synchronous* update Δ_{s} : turmites are simultaneously updated at each time step.

• *cyclic* update Δ_{c}: turmites are updated sequentially in a fixed order.

• *random* update Δ_{r}: turmites are updated sequentially but the order of the updating within each cycle varies randomly.

These updating method are formally defined with, $\forall t \in \mathbb{N}$, :

$$\begin{aligned}
&\Delta_{\mathrm{s}}(t) = \mathrm{T},\\
&\Delta_{\mathrm{c}}(t) = \{t \bmod N + 1\}, \text{ and}\\
&\Delta_{\mathrm{r}}(t) = \pi_k(t'),
\end{aligned}$$

where $t' = t \mod N$, $k = \lfloor t/N \rfloor$ and $(\pi_k)_{k\in\mathbb{N}}$ is a series of independent random variables that draw a single element uniformly in the set of all permutations of T.

Now that we have defined Δ, let us 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 as follows:

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

This rule means that $S_c(t)$, the state of a cell c at time t, is changed if the cell c is selected by Δ and contains at least one turmite. Other policies are possible, for instance considering the so-called "annihilation policy" where simultaneous flips are combined by pairs [6].

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. 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 without tacking into account the other turmites influence in case of conflict.

Let us denote by `R` and `L` the right and left rotations, respectively, such that `R(N)` = `E`, `L(N)` = `W`,... The functions $\tilde{O}_i(t)$ and $\tilde{P}_i(t)$ are defined by:

$$\tilde{O}_i(t) = \begin{cases} \mathtt{R}(O_i(t)) & \text{if } S_{P_i(t)}(t) = 0\\ \mathtt{L}(O_i(t)) & \text{if } S_{P_i(t)}(t) = 1\end{cases}$$

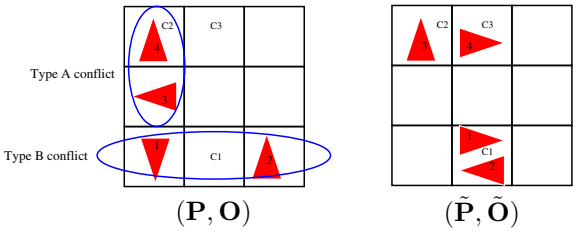


Figure 2: From the current position and orientation $(\mathbf{P}, \mathbf{O})$ of the turmites to their expected next position and orientation $(\tilde{\mathbf{P}}, \tilde{\mathbf{O}})$. We have: $n[\mathbf{P}, c_1] = 0$, $n[\tilde{\mathbf{P}}, c_1] = 2$, $n[\mathbf{P}, c_2] = 1$, $n[\tilde{\mathbf{P}}, c_2] = 1$, $n[\mathbf{P}, c_3] = 0$ and $n[\tilde{\mathbf{P}}, c_3] = 1$. This situation generates a type A conflict with turmite 3 and a type B conflict with turmites 1 and 2 and no conflict with turmite 4.

and

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

where $\mathrm{dP}(\mathtt{N}) = (0, 1)$, $\mathrm{dP}(\mathtt{E}) = (1, 0)$, $\mathrm{dP}(\mathtt{S}) = (0, -1)$, $\mathrm{dP}(\mathtt{W}) = (-1, 0)$ (North, East, South and West translations). We are now in position to define the three different conflict resolution policies ξ.

Allow policy ***(ξ_{AL})***. This policy allows turmites to move and rotate freely without taking into account conflicts. Formally:

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

Exclude policy ***(ξ_{EX})***. When a conflict occurs, the ξ_{EX} prohibit turmites' moves and rotations. 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 turmite positions $\mathbf{P}$:

$$\begin{array}{rccl} n: & \mathcal{L}^N \times \mathcal{L} & \rightarrow & \mathbb{N} \\ & (\mathbf{P}, c) & \mapsto & \mathrm{card}\{i \in \mathrm{T}, P_i = c\}. \end{array}$$

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.

Formally, we say that a turmite i is in a *type A conflict* if $n[\mathbf{P}, \tilde{P}_i] \neq 0$. Similarly, a turmite i is in a type B conflict if: $\exists j, i \neq j, \tilde{P}_i = \tilde{P}_j$, and $n[\mathbf{P}, \tilde{P}_i] = 0$. The `Exclude` policy then writes:

$$\xi_{\mathrm{EX}} : (P_i, O_i)(t+1) = \begin{cases} (\tilde{P}_i, \tilde{O}_i)(t) & \text{if turmite } i \text{ is not in conflict,} \\ (P_i, O_i)(t) & \text{otherwise.} \end{cases}$$

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:

$$\xi_{\mathrm{TS}} : \begin{cases} P_i(t+1) = \begin{cases} \tilde{P}_i(t) & \text{if turmite } i \text{ is not in conflict,} \\ P_i(t) & \text{otherwise,} \end{cases} \\ \\ O_i(t+1) = \tilde{O}_i(t). \end{cases}$$

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. 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, that is, if it qualitatively modifies the orbits. We can now investigate which are the interesting collective phenomena that can be observed with the different submodels.

3 Observations of collective phenomena and their robustness

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 collective phenomena. As an exhaustive exploration of these phenomena is out of reach, it is necessary to select a few phenomena to study. In this section, we select a few collective phenomena that, in our view, can not be predicted simply by looking at the local rules that define the system. Note that we are not only interested in studying this phenomena but we also want to know how they are affected by changes in the simulation schemes (updating method and conflict resolution policies).

3.1 Cycles and Clocks

The first phenomenon that caught the attention of researchers was that a mono-turmite system starting from an all-0 grid leads to the construction of a *path* structure, that is, 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, it is known that it is impossible to observe a mono-turmite whose behaviour is cyclic [4, 8].

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 [7]. However, it is in general difficult to predict the form of a cycle as a function of the initial condition. We

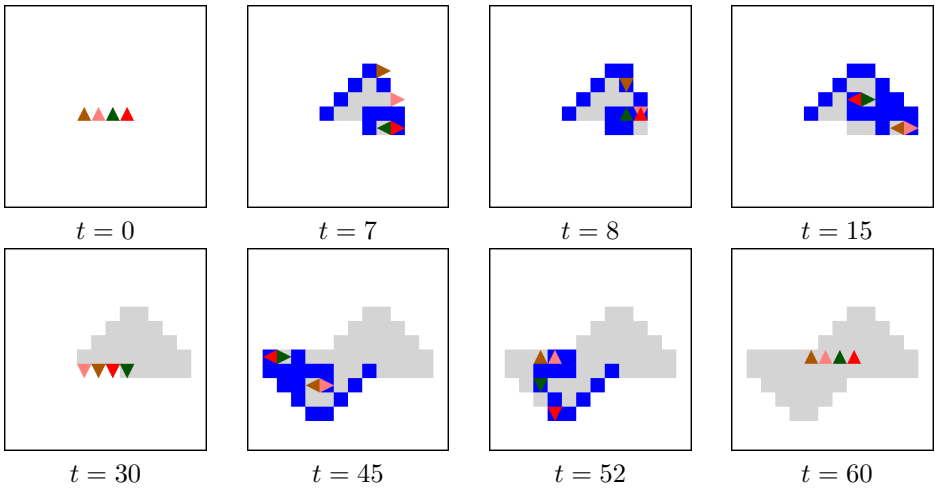


Figure 3: Clock cycle with 4 turmites. Each turmite has its own colour (convention kept).

now present a particular set of initial conditions for which this prediction is possible, forming a phenomenon that we call a *clock*.

Definition 1 (Cycle) *An orbit* $(\sigma(t))_{t\in\mathbb{N}}$ *is a* cycle *if*

$$\exists\, t_0, p \in \mathbb{N},\ \forall t \geq t_0,\ \sigma(t+p) = \sigma(t).$$

The smallest t_0 and p for which the cyclic property is verified are called the *transient time* and the *period* of a cycle, respectively.

Observation 1 *For the synchronous* `Allow` *submodel, an even number of turmites placed horizontally next to each other with a North orientation produces a cycle. Formally: for* $N \in 2\mathbb{N}$*, and for an initial condition* $\sigma = (\mathbf{S}, \mathbf{P}, \mathbf{O}) : \mathbf{S} = \mathbf{0}; \forall i \in \mathrm{T},\ P_i = (i-1, 0), O_i = \texttt{N}$*, the orbit* $\mathrm{ORB}(\Delta_\mathrm{s}, \xi_\mathrm{AL}, \sigma)$ *is a cycle.*

We experimentally determined that the period of the cycle varies as $16N - 4$ and that the transient time is 0. Moreover, the sets of cells visited by the turmites can be enclosed in a rectangular zone of $3N \times 2N$ cells.

Figure 3 shows a cycle with 4 turmites. The configuration at time $t = 60$ shows the end of the cycle, that is, when the configuration is identical to the initial configuration.

Robustness For the set of initial conditions described above, the clock phenomenon is only observed with the synchronous update and the `Allow` policy ($\Gamma_{\Delta_\mathrm{s},\xi_\mathrm{AL}}$ submodel, see Tab. 1). For the 8 other submodels considered, no regularity of behaviour was observed, at least for the first few hundred steps of evolution. The divergence between the evolution of the different submodels is observed only after a few steps.

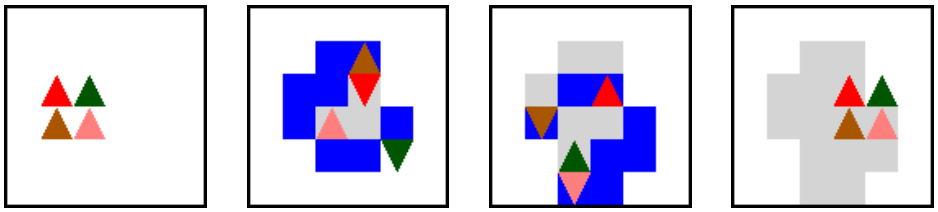


Figure 4: Four steps in the translation cycle of the B-glider.

3.2 Gliders

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

Definition 2 (Gliders) *We say that an orbit* $(\sigma(t))_{t\in\mathbb{N}}$ *is a glider if:*

- *all the turmites are in the same infinite translation, that is, for all* $i \in \mathrm{T}$ *:*

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

- *starting from an empty grid, the "trace" left by the turmites is 0, that is:*

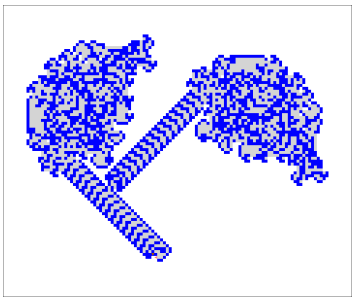
$$\mathbf{S} = \mathbf{0} \quad \text{and} \quad \exists \tau, \forall t, \forall i \in \mathrm{T},\ S_{P_i(t)}(t+\tau) = 0\,.$$

These gliders are phenomena which are analogous to the gliders observed in the Game of life cellular automaton [1]. The first glider was the *F-glider* [7] ; we now present the *B-glider*, described below:

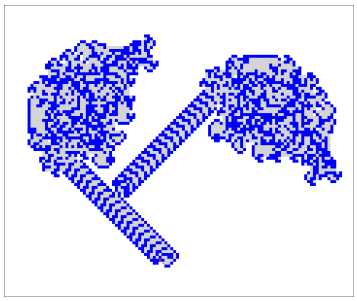
Observation 2 *For the* `Allow` *policy, four turmites placed in a square position with a North orientation form a glider. Formally: for* $N = 4$*, for* $\Delta \in \{\Delta_{\mathrm{s}}, \Delta_{\mathrm{c}}, \Delta_{\mathrm{r}}\}$*, the orbit* $\mathrm{ORB}(\Delta, \xi_{\mathrm{AL}}, \sigma)$ *obtained with the initial condition* $\sigma = (\mathbf{S}, \mathbf{P}, \mathbf{O}) : \mathbf{S} = \mathbf{0}; P_0 = (0,0), P_1 = (1,0), P_2 = (1,1), P_3 = (0,1); O_1 = O_2 = O_3 = O_4 = N\}$ *is a glider.*

Figure 4 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 The B-glider is robust to changes in the updating method. However, it is not robust to changes in the conflict resolution policy (see Table 1). Note that, by contrast, the F-glider is only observed with the synchronous updating and the `Allow` policy.



$k = 52, t = 11539$



$k = 56, t = 11643$

Figure 5: Two stalemate situations obtained with 52 and 56 initial interspace between the two turmites (see text for a precise description of the initial condition).

3.3 Stalemate

We now present our third class of initial conditions, which produces a phenomenon that we call a *stalemate*[2]:

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

$$\exists t_0, \forall t \geq t_0, \forall i \in \mathrm{T},\ P_i(t) = P_i(t_0)\,.$$

The smallest t_0 for which the property is verified is called the *stalemate time*.

Observation 3 *For the synchronous* `Turn'n'See` *submodel, two turmites placed at a horizontal distance k of each other, with orthogonal orientations West and North, always produce a stalemate if k is a multiple of 4 and greater than 52. Formally: for $N = 2$, for $k \in 4\mathbb{N}^*$ and $k \geq 52$ and $\sigma = \{\mathbf{S} = \mathbf{0}, P_1 = (0,0), O_1 = \mathtt{W}; P_2 = (k,0), O_2 = \mathtt{N}\}$, the orbit* $\mathrm{ORB}(\Delta_\mathrm{s}, \xi_\mathrm{TS}, \sigma)$ *is a stalemate.*

Figure 5 shows two stalemates obtained with two distances k. Note that although stalemates are also observed for $k < 52$, there is no regularity in the way the phenomenon happens. By contrast, for $k \geq 52$, we can guarantee that the two turmites always interact in the same way. Indeed, this "minimal security distance" is given to ensure that each turmite does generate its own path and is not "perturbed" by an overlap in 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, turns around it and then catches up with turmite 1. This produces a type B conflict which always results in a stalemate.

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 $(2, -2)$ (see Fig. 5). We also observe that an increase of k by 4, increases

[2]It was originally called a *deadlock*.

the stalemate time by 104, a result in agreement with the analytical results obtained by Boon [3]. The stalemate time thus varies with k as: $t_{\mathrm{st}}(k) = t_{\mathrm{st}}(52) + \frac{k-52}{4} \times 104$, with the experimental value $t_{\mathrm{st}}(52) = 11539$.

Robustness. Stalemates were observed only with the synchronous `Turn'n'See` submodel ($\Gamma_{\Delta_{\mathrm{s}},\xi_{\mathrm{TS}}}$, see Tab. 1). The sensitivity of this phenomenon to asynchrony is analysed in the next section.

To sum up, we described three phenomena that result from the collaboration of turmites. We observed that, depending on the phenomenon considered, there exist a wide variety of responses of the system to the variations of the updating scheme and conflict resolution policy. In the next section, we endeavour to explain some of these variations of behaviour by means of microscopic analysis.

4 Microscopic analysis of the robustness

To understand how the variation from one submodel to another affects the global behaviour of the system, our method consists in examining the evolution of the system until we identify the time steps where a spatial conflict appears. We then try to establish a relationship between the type of collision and the robustness of the system from a global point of view.

4.1 Sensitivity of the Clock

Recall that the "clock" is a collaborative phenomenon that was observed only with the synchronous `Allow` policy (see the previous section). Let us explain this sensitivity to asynchrony and to the changes of conflict resolution policy.

By observing the evolution of the initial condition that generates a clock with the synchronous `Allow` policy, we remarked that:

(a) If the `Turn'n'See` or `Exclude` policies are used, a divergence with the `Allow` policy appears after only one step since the movements of all the turmites but the rightmost one are blocked by these two policies.

(b) When using an asynchronous `Allow` policy, the divergence with synchrony appears later, at a time that depends on the number of turmites involved. For instance, for four turmites, it appears after 8 steps.

To identify the origin of the sensitivity, we observed the orbit of the clock and noticed that it contains a particular type of spatial conflict, that we call the *break* conflict. This conflict is characterised by a particular configuration where two turmites are sharing the same target cell and have the same direction. It is visible on Fig. 3 at time $t = 8$.

As seen on Fig. 6 ($t = 1$), when a break conflict appears, the two turmites leave the cell with opposite directions in the synchronous mode while they leave it with identical directions in the asynchronous mode. The presence of this conflict thus explains the sensitivity of this pattern to variations

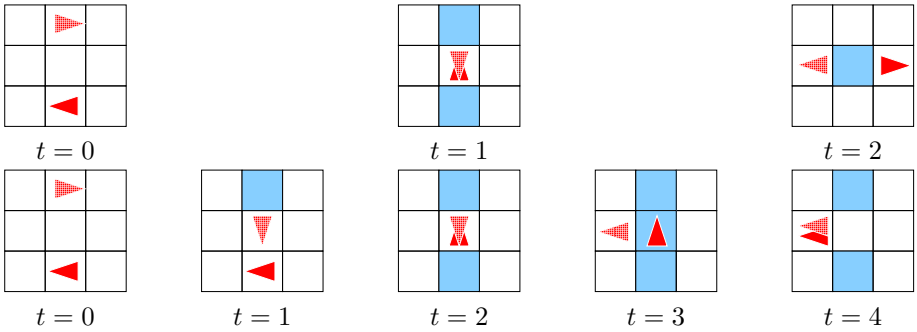


Figure 6: Microscopic analysis showing the non-robustness of the "break" conflict (cells in state 1 are in blue). `Allow` policy with: (top) synchronous, (bottom) asynchronous update. The hatched turmite is updated before the plain turmite (convention kept in the following figures).

of temporal updates: a local divergence in trajectories appears and this small divergence is amplified until it generates qualitatively different orbits (see Tab. 1 page 14). Interestingly, it is also the existence of a similar conflict that explains the sensitivity of the F-glider. Let us now examine a pattern whose response to asynchronism is radically different.

4.2 B-glider

Contrarily to the F-glider, the B-glider is robust to changes in the updating scheme, but not to changes in the spatial conflict policy. An analysis of its different steps of evolution shows that involves only one form of conflict. This conflict appears twice during the cycle: on Fig. 4, it appears at time $t = 4$ and time $t = 8$.

We call this particular form of conflict the *reversion conflict*. For two turmites i and j, it is characterised by the pattern represented in Fig. 7: $O_i = O_j$, $P_j = P_i + (1, 0)$, $S_i = 0, S_j = 1$ (the other patterns obtained by translations and 90° rotations are of course equivalent).

As this is a particular case of type A conflict, the `Allow` policy allows the turmites to swap their positions. With the asynchronous update, the exchange happens in two steps but the result is the same as in the synchronous case, whatever the updating order of the turmites (see Fig. 7). This similarity of evolution explains the robustness of the glider to the asynchronous update.

On the other hand, the `Turn'n'See` and `Exclude` 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. Remarkably, this conflict generally leads to the production of cyclic or translating orbits. Indeed, when it appears, its effect results in the inversion of the roles of the two turmites. Once the conflict has occurred, we generally observe that each turmite erases the trace left by the other turmite. This phenomenon has

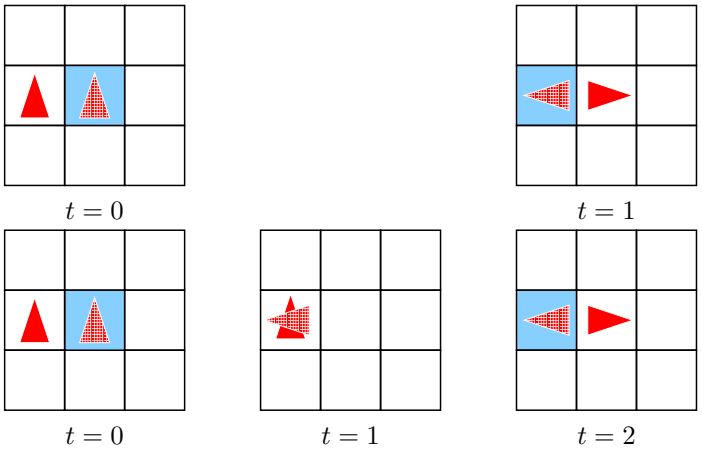


Figure 7: Microscopic analysis showing the robustness of the "reversion" conflict. Allow policy with: (top) synchronous update, (bottom) asynchronous update.

already been observed by other authors (e.g., Chopard and Droz [8], Chevrier and Fatès [7]) but was only partly explained.

4.3 Stalemates

The occurrence of stalemates is characterised by a situation where turmites are always in a type B conflict with perpendicular orientations. We call this type of conflict the *dual lock pattern* (see Fig. 8). Recall that stalemates were observed only with the `Turn'n'See` policy and with a synchronous update.

To explain why, let us consider two turmites i and j such that: $P_j = P_i + (1, -1)$, $O_i = \texttt{E}$, $O_j = \texttt{N}$, $S_i = 0$, $S_j = 1$. As a type B conflict occurs on the cell $c = \tilde{P}_i = \tilde{P}_j$, the new orientations and positions are: $O'_i = \texttt{S}$, $O'_j = \texttt{W}$ and $P'_i = P_i$, $P'_j = P_j$ (the positions of the turmites are unchanged). Then, the same type of conflict appears on the cell $c' = \tilde{P}'_i = \tilde{P}'_j$ and the turmites are again in a type B conflict that results in the stalemate.

On the contrary, with an asynchronous updating, turmite i moves before turmite j, which "frees" the stalemate (see Fig. 8).

Clearly, no stalemate can occur with an `Allow` policy as turmites are not "blocked" by conflicts. With the `Exclude` policy, 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 explains why the stalemate phenomenon was observed only with the synchronous `Turn'n'See` submodel (results reported in Tab. 1). In general, we observed that orbits that involve type B conflicts are not robust to the asynchronous updating.

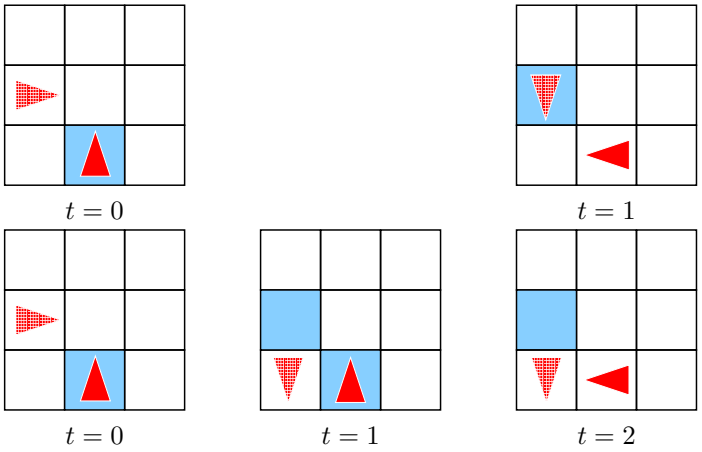


Figure 8: Microscopic analysis of the "dual lock" conflict. Turn and see policy with: (top) synchronous update, (bottom) asynchronous update.

Table 1: Observation of the three phenomena and the three conflict forms that occur in their evolution. Only the submodels that do produce a phenomenon appear in the table.

Phenomena	Submodels	Conflicts
clock	$\Gamma_{\Delta_s,\xi_{AL}}$	break
B-glider	$\Gamma_{\Delta_s,\xi_{AL}}$ $\Gamma_{\Delta_c,\xi_{AL}}$ $\Gamma_{\Delta_r,\xi_{AL}}$	reversion
stalemate	$\Gamma_{\Delta_s,\xi_{TS}}$	dual lock

5 Discussion

We presented clocks, gliders and stalemates as three emergent phenomena in a multi-agent system composed of turmites which evolve on an infinite square grid. Their robustness was tested with nine different simulation schemes. 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.

The formalism we employed allowed us to define various types of orbits and thus to give a rigorous — although partial — definition of the robustness of those phenomena to asynchrony. We exhibited a 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. 1 for a synthesis).

Although the classical Langton's ant (mono-turmite) system is still not fully understood, the multi-turmite system opens an even wider realm to discover. Many other puzzling phenomena deserve to be studied, for instance, the so-called *ever growing square* (or diamond) where the turmites collaborate to produce a pattern that progressively expands (see [18, 7]). The possibility to use turmites to generate textures is another possible direction of research.

The relationship that we noticed between the initial condition, the conflicts' forms and the global behaviour of the system also deserves further analysis. A challenging problem is 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.) We already know from the work of Gajardo *et al.* that a singled-agent system is Turing-universal [13]. We ask how to relate the sensitivity of this system to simulation conditions to its computation universality. Is it the most important feature to understand this system or are there some other key notions that needs to be discovered? To date, it is a challenging problem to find a simple multi-turmite system that would be robust to multiple variations of its simulation scheme (a property that is not verified by the construction of a universal machine in the *Game of Life* [1]). We believe that a deeper understanding of clocks, gliders, stalemates as well as other collaborative phenomena may provide some hints to answer these questions.

Finally, we ask how the investigations made on the robustness of the multi-turmite system can be related to other types of complex systems. In particular, it would be interesting to compare the robustness of our model with the lattice-gas cellular automata models of Chopard and Droz [8] or to other models where the interactions between cells and turmites have a physical interpretation. As the qualitative behaviour of a system may highly depend on small simulations details, we need to develop specific tools to understand how interactions generate robust or sensitive collaborative phenomena. Is the sensitivity to the simulation scheme observed here a rare or a common phenomenon ? Can it be seen only in

simple, discrete and deterministic models or can it also be observed in a wider range of models?

References

[1] Elwyn R. Berlekamp, John H. Conway, and Richard K. Guy. *Winning Ways for your Mathematical Plays*, volume 2. Academic Press, ISBN 0-12-091152-3, 1982. chapter 25.

[2] Olivier Beuret and Marco Tomassini. Behaviour of multiple generalized Langton's ants. In C. Langton and K. Shimohara, editors, *Proceeding of the Artificial Life V Conference*, pages 45–50, Nava, Japan, 1998. MIT Press.

[3] Jean Pierre Boon. How fast does Langton's ant move? *Journal of Statistical Physics*, 102:355–360, 2001.

[4] Leonid A. Bunimovich and Serge E. Troubetzkoy. Recurrence properties of Lorentz lattice gas cellular automata. *Journal of Statistical Physics*, 67:289–302, 1992.

[5] Buvel, R.L. and Ingerson, T.E. Structure in asynchronous cellular automata. *Physica D*, 1:59–68, 1984.

[6] Vincent Chevrier and Nazim Fatès. Multi-agent Systems as Discrete Dynamical Systems: Influences and Reactions as a Modelling Principle. Research report, INRIA - LORIA, 2008.

[7] Vincent Chevrier and Nazim Fatès. How important are updating schemes in multi-agent systems? an illustration on a multi-turmite model. In *Proceedings of the 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS), Toronto, Canada*, pages 533–540, 2010.

[8] Bastien Chopard and Michel Droz. Cellular automata modeling of physical systems. pages 46–51. Cambridge University Press, 1998.

[9] Alexander K. Dewdney. Two-dimensional Turing machines and tur-mites make tracks on a plane, computer recreations. *Scientific American*, pages 180–182, Sept. 1989.

[10] Nazim Fatès. Does *life* resist asynchrony? In Andrew Adamatzky, editor, *Game of Life Cellular Automata*, pages 257–274. Springer London, 2010.

[11] Nazim Fatès and Lucas Gerin. Examples of fast and slow convergence of 2D asynchronous cellular systems. *Journal of Cellular Automata*, 4(4):323–337, 2009.

[12] J. Ferber and J.P. Müller. Influences and reaction : a model of situated multiagent systems. In *Proceedings of the 2nd International Conference on Multi-agent Systems*, pages 72–79, 1996.

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

[14] Anahí Gajardo and Eric Goles. Dynamics of a class of ants on a one-dimensional lattice. *Theoretical Computer Science*, 322(2):267–283, 2004.

[15] Anahí Gajardo, Eric Goles, and Andrés Moreira. Generalized Langton's ant: Dynamical behavior and complexity. In Afonso Ferreira and Horst Reichel, editors, *Proceedings of STACS 2001*, volume 2010 of *Lecture Notes in Computer Science*, pages 259–270. Springer, 2001.

[16] David Gale, Jim Propp, Scott Sutherland, and Serge Troubetzkoy. Further travels with my ant. *Mathematical Entertainments column, Mathematical Intelligencer*, 17:48–56, 1995.

[17] A. Khuong, G. Theraulaz, C. Jost, A. Perna, and J. Gautrais. A computational model of ant nest morphogenesis. In *Advances in Artificial Life, ECAL 2011 - Synthesis and Simulation of Living Systems*, pages 404–411. MIT Press, 2011.

[18] Christopher G. Langton. Studying artificial life with cellular automata. *Physica D*, 22:120–149, 1986.

[19] Damien Regnault, Nicolas Schabanel, and Eric Thierry. Progresses in the analysis of stochastic 2D cellular automata: A study of asynchronous 2D minority. *Theoretical Computer Science*, 410(47-49):4844–4855, 2009.

[20] Alan M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London*, B 237:37–72, 1952.