

DRIMA: A Minimal System for Probing the Dynamics of Change in a Reactive Multi- Agent Setting

Pedro P. B. de Oliveira

DRIMA is a simple cellular model for the multi-agent setting in which reactive agents have their behavior changed by the behavior of others, as the outcome of their interactions; it is also the system that implements the model in *Mathematica*. It was conceived as a metaphor for the high-level issue of how agents “attract” others toward them, be it in the form of a change in any behavioral or conceptual orientation, habit, thinking, etc. This is modeled through a single behavior of the agents, which is their movement on a two-dimensional grid; as they move, they undergo interactions with each other that modify the way they move before and after the interaction. The focus of the model is on addressing issues related to the emergent dynamics of a particular setting, much in tune with an artificial life or complex systems perspective. DRIMA is purposefully meant to be simple and non-general, a minimal system for the kind of question it is designed to help address. The article is a presentation of the model and of key aspects of its implementation, not a discussion on its use to address any particular question.

■ 1. Introduction: The Underlying Motivation

DRIMA is an acronym for **D**ynamics of **R**andomly **I**nteracting **M**oving **A**gents (although, **D**ynamics of **R**andom **I**nteractions in **M**athematica also comes to mind...). DRIMA is a simple model for the multi-agent situation in which reactive agents have their behavior changed by the behavior of others, as the outcome of their interactions. DRIMA is also the system that implements the model in *Mathematica* [1]. Various systems with which DRIMA shares a common architectural, conceptual, and/or motivational basis appear, for example, in Langton [2], Weiss [3], and Bonabeau, Dorigo, and Theraulaz [4], most notably those possessing a cellular nature and following an artificial life perspective.

The only behavior that is embedded in the model is the way the agents move on a two-dimensional grid. And as they move, they undergo interactions with each other that modify the way they move before and after the interaction. A *world* in DRIMA is a snapshot of the state of all the agents of the world, which includes their current grid position and movement pattern, and the values of any other properties that may have been defined for them. On its part, a *history* is the sequence of worlds generated through a pathway of agent interactions, starting from an initial world configuration. Histories are, therefore, enacted, in the sense of Varela, Thompson, and Rosch [5], out of the interactions that couple the whole system, through the common currency of the potential modifications in the individual patterns of movement.

Useful metaphors that act as motivations underlying the model refer to questions related to how the behavior of an agent—say, a consumer, an apprentice, a client, etc.—is affected by the various forces that may pull them in one direction or another; for instance, how a consumer is affected by the choice of products of the same kind. Due to its simplicity, the model only allows you to ask about the overall dynamics of the behavioral/conceptual change of the agent at issue, in face of the various possibilities offered to it; naturally, the model does not support specific questions involving such details as the role of the agent's personality, its world view, its background knowledge on the issue being offered, its cultural bias, and many more aspects. DRIMA is purposefully meant to be simple and non-general, a minimal system for the kind of question it is designed to help address.

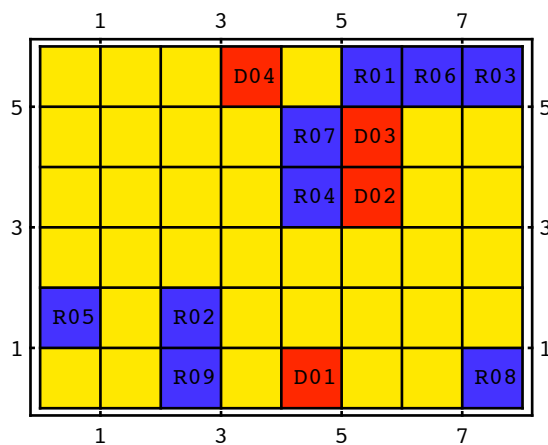
In the remainder of the article, DRIMA is initially described in general terms in the next section. Then a presentation of its implementation is given in Section 3, with a focus on the representation of the agents and the world, as well as on the main aspect of its interactions. In the last section some concluding remarks are made. The reader should be aware that the purpose of this article is only the presentation of the model and of key aspects of its implementation, and not a discussion on its use to address any particular question, for which the reader should refer to Poiani and de Oliveira [6] as a first effort along this line.

■ 2. The Model

□ 2.1 Overview

DRIMA is defined by a discrete, two-dimensional grid and two types of agents that move about in it. Each agent occupies a single cell of the cellular-automaton-type grid, which has periodic boundary conditions; that is, each edge of the grid is wrapped around its opposite edge. The movement of an agent is to one of the eight possible next-neighboring positions of the grid cell where the agent is; no movement may also occur, in which case the agent remains at its original cell. The neighboring positions of a cell are referred to according to the eight points of the compass: East (E), Northeast (NE), North (N), Northwest (NW), West (W), Southwest (SW), South (S), and Southeast (SE). The no-movement situation is referred to as X.

The two types of agents are named D and R, according to their movement style: while D-agents move deterministically, R-agents move randomly. Agents can interact in a binary fashion, that is, only two at a time. The two agents can interact when they get close enough to each other, a distance referred to as *interaction range*.



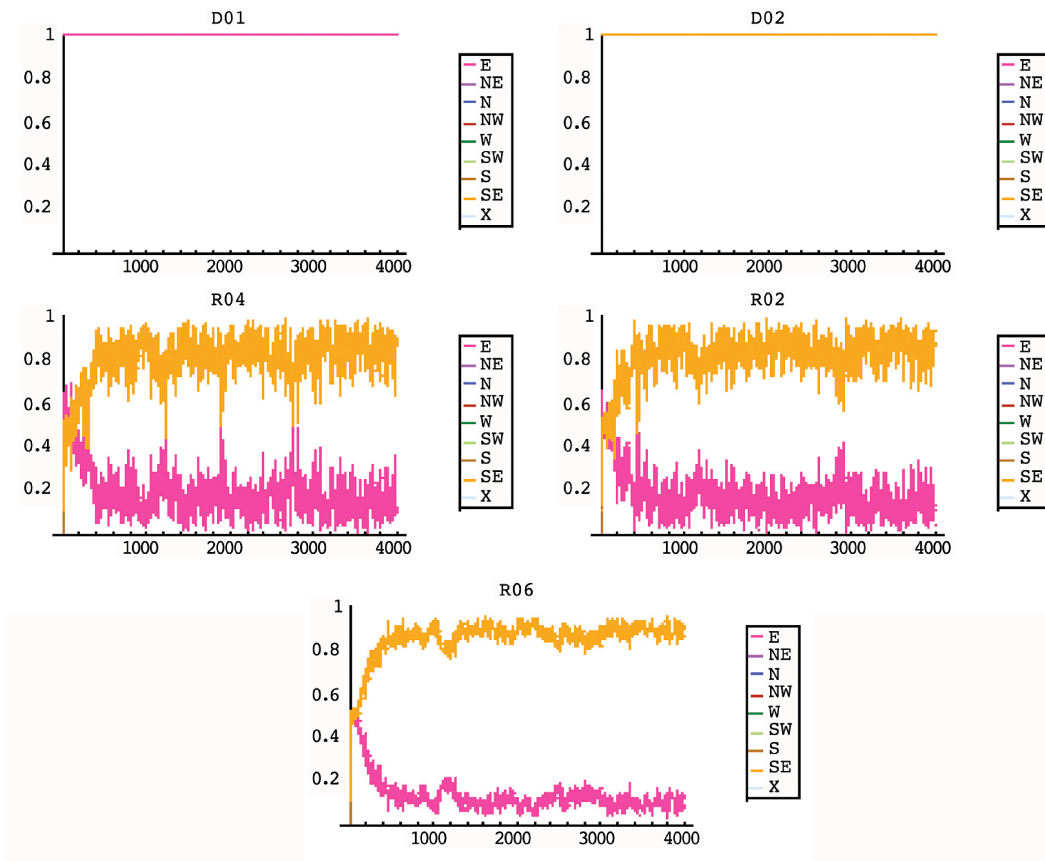
▲ **Figure 1.** Visualization of a DRIMA world. Background is shown in yellow, R-agents in blue, and D-agents in red (with all agents fully tagged). This is a 6×8 world, with nine R-agents and four D-agents, randomly generated and randomly placed.

As mentioned before, the only behavior embedded in the system is the way the agents move, during which they undergo interactions with each other that modify the way they are moving. The agents in the system follow an *interact-first-then-move* cycle, instead of *move-first-then-interact*. Although there does not seem to be any long-term difference between the two approaches, the former was thought to be conceptually more adequate than the latter, since the initial DRIMA world is initialized with all agents already having a rather uniform, predefined initial movement pattern.

The way an agent moves is described by its movement pattern. The movement pattern of a D-agent is such that with probability 1, it moves in a single direction at any time, and with probability 0, it moves in all the other directions. On the other hand, the movement pattern of an R-agent is such that different probabilities define the bias of its movement in any direction, including the chance that it does not move at all; naturally, these probabilities must total 1.

The outcome of the interaction between two agents is a change in their individual movement pattern. The idea is that an agent that moves in a certain direction with smaller probability is attracted by the other agent in that direction, through the smaller probability being slightly increased. For instance, if there are a single D-agent and various R-agents in the world, the patterns of movement of the R-agents become more and more deterministic in the direction of the D-agent, such that, in the long run, all of them end up with the same movement pattern of the D-agent. In other words, the R-agents have been attracted by the D-agent, up to their converging to the actual way the D-agent moves. Convergence like this is not always the case, though. Even in the case of only two D-agents moving in different directions, the various R-agents may not be able to settle down to any stable movement pattern, and very complicated dynamics may come about. The details of how the interaction is actually implemented are given below, together with many more details.

Figure 2 displays the evolution of the movement patterns of agents in an 8-column by 5-row world, composed of two D-agents and six R-agents, during a history of 4000 worlds. Each display plots the history of probability values that govern the ability of the corresponding agent to move in every possible direction. The top two figures refer to the D-agents, each one deterministically moving in a different direction, SE and E, and at the bottom, the graphs refer to three of the six possible R-agents.



▲ **Figure 2.** Evolution of the movement patterns of agents of an 8×5 world composed of two D-agents and six R-agents, with the D-agents moving in different directions, during a history of 4000 worlds. Only the movements of three R-agents are on display.

Although the figure shows that the D-agent's E-movement has the prevailing influence in the history, the display sequence suggests the details are distinct from each other, depending on the R-agent one looks at. In fact, the sequence was chosen so as to make clear a progression in the degree of influence of the D-agents over the R-agents. The last R-agent shown, R06, not only rapidly resolves itself in the two available directions, but also does so with determination; that is, its average probability of moving east quickly goes up, and remains so with little variance (naturally, its probability of moving southeast quickly goes down, also with little variance). This is in contrast with R04, the first R-agent shown, which seems to resolve itself much more slowly and with much less determination. In its evolution it is clear that, up to around the first 200 worlds, the southeast direction seems to be prevailing; then, from that stage onward, the east direction starts picking up, until reaching the average probability level that is kept afterward, even though with a large

variance. But notice that even after the east direction has picked up, all of sudden downward bursts can be seen, evidences of a fluctuation in the agent's determination to continue moving primarily to the east. Following R04 in the display sequence, R02 is on display, in which the frequency of those downward probability bursts toward the southeast has decreased, eventually reaching a stage where they disappear completely, as happens for the subsequent agent, R06.

□ 2.2 Dimensions of the Interactions

Whenever interactions are involved in a computational system, various issues have to be accounted for in order to define the nature of the interactions allowed in the system. These issues can be regarded as *dimensions* that define a space of possible interactions. Among these dimensions, the following are the ones that were considered when setting the features of DRIMA.

1. **Range of Action:** *Moore* neighborhood of range (radius) R . This means that the range of action of the agents is fully isotropic and there is no preferential direction; in other words, the agents influence their surroundings in a fully symmetrical way, regardless of the direction, and with the same intensity in all of them. Although any value of the range of action R can be specified in the system, usually it is equal to 1 (a single cell position), which means that an interaction may happen when two agents are at contiguous grid locations. If the interaction range is set to zero, this means that interactions can only occur when both agents are sharing the same grid cell.
2. **Scope:** *Binary* or *N-ary* (multiple). In the current definition/implementation of the system, only binary interactions are allowed; that is, only two agents are considered at any interaction, even if more agents are present in the range of action of an agent. Binary interactions were chosen so as to keep the system simpler. Multiple agent interactions should not be totally discarded in the future, though.
3. **Type:** Interactions involving agents of the *same* kind, or of *different* kinds. All useful kinds of interactions can be defined by the user, in particular by specifying for each one the type of agent that should be affected or not; that is, which type of agent should come out of the interaction changed or unchanged.
4. **Mode:** *Synchronous* or *Asynchronous*. Both are implemented in the system and are related to the way the world configuration is updated during a history. In the asynchronous mode, an agent is randomly picked from the world, together with one of its neighbors, also randomly selected, and the interaction happens. In contrast, in the synchronous mode every agent in the world is sequentially considered for interaction, but only those that have not yet been processed in the current iteration, that is, those that have not yet been selected as a neighboring agent of a previous interaction. For each agent in the sequence, one of its neighbors is randomly drawn, and the interaction is carried out. In order to avoid artifacts in the sequence of agents chosen to interact, a random permutation of the agents is taken at every iteration.

5. **Direction:** *One-way* or *Two-way*. As the outcome of an interaction, both agents may be affected, none of them, or only one of them. So, while in the case of one-way interactions only one of the agents involved is affected, in the case of two-way interactions, the two agents can be affected. In both cases, the agent that should not take part in the interaction simply comes out of it unchanged (presently, only one-way interactions have been implemented). In DRIMA this dimension is specified by the user at the level of the (eight possible) individual directions of movement of an agent, not at the level of the agent as a whole; this has consequences that will be explained in Section 3.2.
6. **Effect:** *Direct* or *Indirect*. The aim of the interactions is to change the way agents move. One way this may happen is directly, in which case the patterns of movement of an agent just before and after the interaction are distinct from each other. However, the changes may not be allowed to come into effect immediately, being mediated by some feature of the agents that would allow a change in an agent's movement pattern to be triggered only when reaching a certain value. This is like an energy-type feature that first has to be accumulated by an agent, or a barrier that has to be overcome by it, before its movement pattern is allowed to change. Naturally, in this situation, the variation of such an agent property is the real, foreground outcome of the interactions, and therefore the consequence of the interactions upon the agents' movements becomes an indirect effect. In DRIMA only direct effects of the interactions are fully implemented, and therefore only those are presented in this article.
7. **Intensity:** The amount of variation in what is changed at the interactions. In the system, *fixed* or *variable* amounts of variation are possible and, for each, the possibility of setting the actual value from a set of discrete alternatives exists. So, in the direct kind of effect, the agents can have their probability of movement toward a certain direction altered by a fixed or variable amount as they participate in an interaction, where the actual value involved is chosen from a predefined set of possibilities.
8. **Determinism:** *Deterministic* or *Non-Deterministic*. Although agents move in non-deterministic fashion, the interactions implemented in DRIMA are usually deterministic. However, this is not necessarily so, and in fact, some non-deterministic interactions are implemented in the context of indirect interactions, where the intensity of the interaction for the agents involved is constrained but randomly chosen.
9. **Content:** *Reactive* or *Cognitive*. The interactions in DRIMA are fully reactive, in the sense that their outcome depends only upon the current state of the agents involved, very much like finite-state machines [7]. Whatever the role of the agents' past histories, they are compressed, so to speak, in their current state, and this is the single point that matters. Hence, the agents have no internal representations of the world and no explicit memories of their past, and therefore no interaction in the system is content rich, as usually happens with cognitive agents.

■ 3. Implementation

In general terms DRIMA's implementation follows a fully functional programming style, and relies on `DiscreteMath`Combinatorica``, `Graphics`Graphics`` and `Graphics`MultipleListPlot`` packages. (These packages are now obsolete.) Usage of the systems assumes that a world is initially created with a set of agents and all their properties and initial positions, followed by a sequence of world updates, from which the agents move about and interact with each other. The updates stop when a given history length (number of updates) is achieved. At the end, the world history is ready for analysis, directly available in the *Mathematica* session in use or from a log file.

In order to facilitate the latter, a few functions are available for visualization, most notably `DisplayWorld[world]` and `DisplayHistory[history]`. The main functions controlling the system's dynamics are `SyncWorldUpdate[world, worldproperties]` and `AsyncWorldUpdate[world, worldproperties]`, which perform a single synchronous or asynchronous update of the world, respectively, according to what is required in a run. Notice that since these world update functions implement a single update, they have to be encapsulated into iterative functional or imperative *Mathematica* structures (like `Fold` or `Nest` in the first case, and `Do` or `For` in the second) in order for a history to be created. In addition to the latter two types of functions (world update and visualization), various others are also available in the system for input-output, initialization of the world, general handling of the agents' characteristics, movement control of the agents, direct and indirect agent interactions, handling of the supported neighborhoods where interactions are allowed to happen, convergence analysis of the agents, and miscellaneous utilities. In total, there are currently 49 functions, grouped into the 11 types just mentioned.

In the next two sections a detailed presentation is given of the representation of the agents and the world, and of the most important form of interaction (the direct one) presently in use. These details are required for a proper understanding of the system's structure, and so that this article becomes self-contained in terms of how DRIMA operates from the perspective of a potential user.

□ 3.1 Agent and World Representation

Every agent in DRIMA is represented by the list $\{identity, position\}$, where *identity* is the set of features that define the agent's name and state, and $position = \{x, y\}$ gives the agent's current coordinates in the world.

identity is the list $\{name, movement\ pattern, list\ of\ properties\}$, where:

- *name* is a string formed by the concatenation of the agent type (D or R) and a number to identify it uniquely.
- *movement pattern* = $\{P_E, P_{NE}, P_N, P_{NW}, P_W, P_{SW}, P_S, P_{SE}, X\}$, where P_d is the probability that an agent will move in the d -direction (X representing the no-move probability, i.e., the probability of its staying at the same grid location).
- *list of properties* may contain additional features that might be ascribed to the agents in a particular setting. For instance, one could define features like the agent's susceptibility to change in an interaction, in the sense that the larger its susceptibility, the larger the intensity of an interaction on its movement pattern; analogously, one could define the agent's *influence potential*, which could be thought of as its power to drive the intensity of an interaction, so that the larger this feature in an agent, the larger the intensity of its interactions over the agent it interacts with.

DRIMA represents the world with the list $\{\textit{list of agents}, \textit{grid size}, \textit{types of interaction}, \textit{world mobility}\}$, where *list of agents* is simply a list with all agents in the world, defined as above, and *grid size* = $\{X_{max}, Y_{max}\}$ is the size of the grid that defines the world space.

world mobility is a user-defined parameter that can take on either F or P, standing for, respectively, full or partial mobility. If full mobility is specified, all agents are required to keep moving at every iteration, that is, their no-move probability is always kept at zero. On the other hand, partial mobility entails that the ability of an agent to stay at the same grid location be handled just like the probability of the agent moving in any direction.

interaction types is a list of the types of interaction allowed in the world, the possibilities being $\{\text{dd}, \text{DD}, \text{rr}, \text{RR}, \text{dR}, \text{Rd}\}$. Since there are two types of agents, one might assume that there would be only four types of interaction; however, the notation in use, which distinguishes capital from lowercase letters, extends the number of alternatives. If the agent type is specified using uppercase letters, this means it will undergo an effective interaction; that is, its movement pattern is allowed to come out changed from the interaction. Conversely, if the agent type is specified using lowercase letters, the interaction becomes ineffective to the agent, as it will always come out unchanged. For instance, while DD entails that both D-agents can be modified by the interaction, dd entails they never will be. And although interactions Dr or rD could have, in principle, been defined, they were discarded because they do not seem to represent a useful situation in DRIMA.

The default interactions are $\{\text{dd}, \text{RR}, \text{dR}, \text{Rd}\}$, and the specification $\{\}$ entails no interaction is allowed in the world, so that the agents simply roam around, without ever having their movement patterns altered. The interaction pairs DD – dd or RR – rr should not be specified, as they conflict (which is resolved by assuming that DD and RR will prevail in each pair, respectively). Finally, for an interaction between agents D and R to occur, it suffices to specify dR or Rd.

Examples of the representation of two agents (D0 and R01) are shown below, first in the situation of full (F) world mobility, in which the right-most probability of their movement patterns is set to zero (and will remain so throughout the iterations), and then in the case of partial (P) mobility, where the agents may have a larger-than-zero probability of not moving. But notice in both cases that the sum of all probabilities in an individual movement pattern must total 1. For example, the representations below also show the *list of properties* of an agent, a presently implemented property (a list named *transition potential pattern*), which makes sense only in the context of the energy-like feature of indirect interactions (not discussed here).

F :

```
{{"D01", {0, 0, 0, 1, 0, 0, 0, 0, 0},
      {{0, 0, 0, 1000, 0, 0, 0, 0, 0}}}, {2, 2}}
{"R01", {0.125, 0.125, 0.125, 0.125, 0.125, 0.125,
      0.125, 0.125, 0}, {{0, 0, 0, 0, 0, 0, 0, 0, 0}}}, {4, 6}}
```

P :

```
{{"D01", {0, 0, 0, 0, 0, 0, 0, 0, 1},
      {{0, 0, 0, 1000, 0, 0, 0, 0, 0}}}, {2, 2}}
{"R01", {0.111, 0.111, 0.111, 0.111, 0.111, 0.111,
      0.111, 0.111, 0.111}, {{0, 0, 0, 0, 0, 0, 0, 0, 0}}},
{4, 6}}
```

As far as the world representation is concerned, the following is one possible example.

```
{{{"D01", {0, 0, 0, 1, 0, 0, 0, 0, 0},
      {{0, 0, 0, 1000, 0, 0, 0, 0, 0}}}, {2, 2}},
 {"R01", {0, 0.13, 0.14, 0.11, 0.25, 0.15, 0.10, 0.12, 0},
      {{0, 0, 0, 0, 0, 0, 0, 0, 0}}}, {5, 3}}
 {"R02", {0.125, 0.125, 0.125, 0.125, 0.125, 0.125,
      0.125, 0.125, 0.}, {{0, 0, 0, 0, 0, 0, 0, 0, 0}}},
 {4, 6}}}, {5, 10}, {dd, RR, dR, Rd}, F}
```

When a world is created, all agents are initialized with equal probability in all fields of their movement pattern. The fields of their transition potential pattern are all set to 0, except the one associated with the direction of movement of the D-agents, which is set to a high value, in this case, 1000.

□ 3.2 The Direct Interactions

In order to clarify the details of the interaction process, let us represent the movement patterns of agents i and j by $\{P_E^i, P_{NE}^i, P_N^i, P_{NW}^i, P_W^i, P_{SW}^i, P_S^i, P_{SE}^i, X^i\}$ and $\{P_E^j, P_{NE}^j, P_N^j, P_{NW}^j, P_W^j, P_{SW}^j, P_S^j, P_{SE}^j, X^j\}$, respectively. For instance, the fields P_{NE}^i and P_{NE}^j represent the movement probabilities of agents i and j , respectively, in the northeast direction; or, more generally, P_d^k stands for the movement probability of the k -agent in the d -direction.

What defines the direct interaction between two agents, i and j , are the pairwise probabilities at their corresponding movement pattern fields, P_d^i and P_d^j , for each individual d -direction. In other words, the new movement pattern of an agent that comes out of a direct interaction derives from a local operation involving the pairwise movement pattern fields of the interacting agents. This operation yields new probability values (one or two, depending on whether the interaction is one way or two way) to every movement pattern field of the agents, individually considered, as a function of the current values of the pair. Therefore, it is the local operation at each individual direction field that entails the global consequence of changing the movement pattern of the agent.

When the interaction is meant to be one way, this local operation is simply a probability increase of the smallest probability of the pair, by a predetermined amount Δp ; in other words, this means that only the smallest movement probability of the movement pattern field is affected. But notice that, since the probability increases do happen at the level of each of the individual directions of movement and not at the level of the movement pattern as a whole, while one of the interacting agents may have a probability increase in one direction, the other agent may have a probability increase in another, both increases being the result of the same interaction. This means that while the interaction is one way at the level of the individual directions, it is not necessarily so at the level of the movement pattern as a whole.

In contrast, whenever two-way interactions are referred to, what is really at issue is the fact that movement pattern fields corresponding to the same direction are allowed to change as a direct result of the interaction, thus necessarily entailing that both movement pattern fields are always affected. Although two-way interactions have not yet been implemented in the system, the idea is that while the smallest probability value P_d^i would be increased by Δp^i , the largest P_d^j could decrease by Δp^j (in particular, one might define $\Delta p^i = \Delta p^j$).

In order to preserve the central idea of an interaction (that its outcome should be one agent being *attracted* by the other in terms of the way they move), the probability changes in any given direction must not allow the smallest value to grow larger than the highest probability value before the interaction. This is taken care of in the implementation.

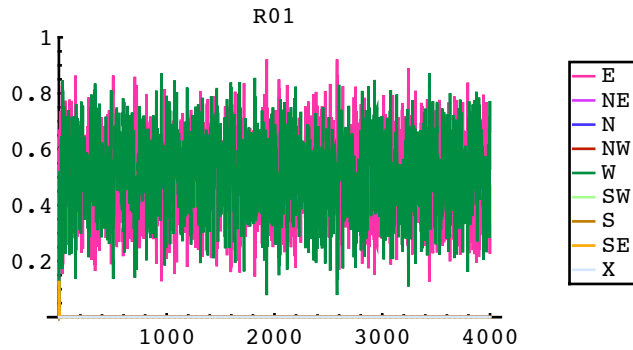
A primary criterion for choosing the Δp value in a certain experiment is to bear in mind its relevance to the richness of the system dynamics. In fact, depending on features like the grid size and the number of agents of each kind, different values of Δp may lead to very distinct qualitative dynamics. In addition, one should also be concerned by the practical consequence that Δp has an impact on the number of interactions required for an agent to go from absolute random behavior (say, 0.11 in all directions, when it is initialized in full mobility) to a practically deterministic movement, when the probability values at all movement pattern fields are all close to 1.

In tune with the latter, the values of probability increase/decrease can be thought of in two ways: a *fixed mode*, meaning that Δp is fixed, regardless of the probability values involved in the movement pattern field at issue; and a *variable mode*, in which Δp is allowed to change proportionally to the actual probability values in the corresponding field. Equations (1) and (2) display the expressions currently implemented:

$$\text{Fixed mode: } \Delta p = \frac{F_p}{9} . \quad (1)$$

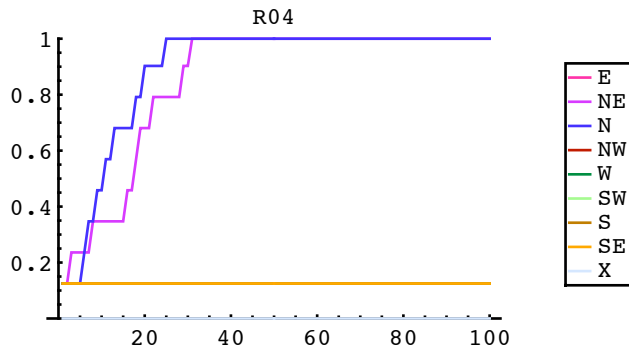
$$\text{Variable mode: } \Delta p = \left| P_d^i - P_d^j \right| \frac{F_p}{9} . \quad (2)$$

As a practical matter, in the experiments in which DRIMA has been tried out and tested [6], a fair combination of parameters has been fixed mode with $F_p = 1$, as the observed dynamics have been shown to be fast in convergence and rich in possibilities. In this situation, $\Delta p = 1/9$ in fixed mode, but also in variable mode when $|P_d^i - P_d^j| = 8/9$, which empirical evidence has shown to be a good value in the experiments performed. It can be seen that with this Δp value, an R-agent and a D-agent would have to interact for about 100 times before the movement behavior of the former converges to that of the latter. Larger values of F_p , say, $F_p = 5$, can speed up convergence (if this is at all possible), but also render it more unstable. Figure 3 exemplifies such an instability with a 4000-world-long history obtained with two D-agents and six R-agents.



▲ **Figure 3.** Example of instability in an R-agent's long-term movement behavior, when a high probability increment factor is used ($F_p = 5$); the figure refers to variable mode, but in fixed mode the same qualitative behavior is observed, usually with even stronger instability.

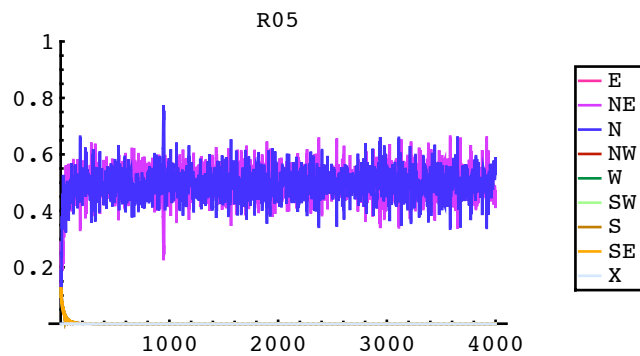
Naturally, since agents are allowed to independently interact at each direction field, individually taken, there is the necessity of normalizing the resulting movement pattern, so as to guarantee the total probability of movement of the agent remains equal to 1. Normalization is also required in order to ensure that R-agents are allowed to become deterministic in the presence of various D-agents. For instance, if both directions defined in a world with two D-agents would individually be accounted for by the R-agents, without normalization, this would entail that their probabilities could become equal to 1 in both directions, which, in practice, represents lack of convergence to either of them. In addition, the actual convergence to 1 would happen too quickly, entailing very uninteresting dynamics. Figure 4 depicts the situation.



▲ **Figure 4.** Lack of normalization entails very uninteresting dynamics and a difficulty for R-agents to become deterministic.

The first normalization scheme that was tried out was the standard way of simply decreasing the excess of 1 among the various direction fields, proportionally to the probability

value at each field. However, this kind of normalization was discarded, as it does not yield rich dynamics. This is because any probability increase at a given direction of movement becomes somewhat flattened out; as a consequence, it becomes very difficult for an R-agent to have its movement differentiated in any given direction. For instance, various 4000-world-long histories obtained with two D-agents and six R-agents all had the same general feature, in which all R-agents settled to a roughly undefined behavior between the two directions represented by the D-agents, with average moving probability close to 0.5, and very little discrimination between any of the two preferential directions. Figure 5 displays a typical situation.



▲ **Figure 5.** Example of the lack of definition in an R-agent's long-term movement behavior, when standard normalization is used.

As an alternative that yielded much richer dynamics, the idea of displacing all probability values by the same amount was then implemented; see Figure 2 for an example. However, two situations have to be distinguished, depending on whether the non-normalized total probability of a movement pattern is larger or smaller than 1. The amount of probability to be subtracted from (or added to) every movement pattern field is given by the division of the excess above 1 (or the amount less than 1) by the number of fields with probability values larger than 0. Then, if any negative probability value comes about, it is made equal to 0. Naturally, whenever the latter happens, normalization can be lost; in this situation, the normalization process is allowed to iterate.

■ 4. Conclusion

DRIMA was conceived as a metaphor for the high-level issue of how agents attract others toward them, be it in the form of a change in any behavioral or conceptual orientation, habit, thinking, etc. On a very different scale, the model also brings forth the idea of a physical particle system, in which random walkers are affected by some force that little by little drives them to more focused patterns of movement that privilege some preferential directions. Regardless of the level of abstraction considered, the role of the non-determinism that permeates the model is simply a realization that avoids artifacts that any deterministically driven choice on the part of the attracted agent might impose. In this sense, the non-determinism condenses any deterministic possibility into a single abstraction. Finally, the role of the explicit notion of space represented by the grid is to couple all activities, thereby constituting yet another simple abstraction for the multitude of possibilities that might be involved in the interrelation between agents, whichever the instantiated metaphor. These are the essence of DRIMA, as a model.

At first glance, the idea of creating a simple, minimal computational model of a multi-agent system, particularly one with a cellular nature, may seem against modern trend in the area [8]. However, at a closer look, the situation is somewhat different. First of all, the usefulness of cellular settings in artificial-life-type models has been proven repeatedly, to the extent that they became very popular in the field, and DRIMA is definitely inspired by that way of thinking (or the complex systems standpoint, for that matter), with its focus on the emergent dynamics of the system. Second, the decision on what kind of model should be used depends upon the kinds of questions one is willing to ask. The issues concerning the dynamics of the changes involved with fully reactive agents whose single role is to move seem simple enough not to demand any fancy sophistication in the model that might blur the focus of what is being asked. And finally, DRIMA is motivated by NKS-type ideas [9], in the sense that it has been a principle in its conception to keep it structurally simple, under the expectation that complex dynamics would be likely to be observed from even simple initial conditions (which, in fact, turned out to be the case). This is implied by the NKS principle of computational equivalence, from which simple models are sufficient to model complex behaviors (see [10] and [11] for recent examples).

But the achieved simplicity in DRIMA might still go further; after all, the present realization of DRIMA is not meant to be the absolute minimal possibility. One alternative for devising similar such systems might be searching through the space of possible related architectures, much in the spirit of [12], in tune with the NKS methodology. But more specifically related to DRIMA's current architecture, one way to go toward pursuing

further simplification would be to simplify its current interaction scheme, from its reliance on the interaction between the two individual corresponding components of the movement patterns of the interacting agents to one that would be based on the actual vectorial resultant of the movement patterns. This would certainly bring more clarity to the whole scheme. In particular, it would allow a clearer account for the indirect interactions mentioned in Section 2.2 which, even though already partially implemented in the system, were excluded from detailed presentation in this article exactly because their present formulation is not yet conceptually satisfactory in terms of simplicity.

Another possibility for further simplifying the model is to create its one-dimensional version. This too has already been performed, even though not discussed herein either; but it is worth mentioning that such a step was very much facilitated by the structure of DRIMA's *Mathematica* code, which allowed its extension quite naturally. The problem with the one-dimensional version is that the present version allows for a clearer account of the agents' behaviors, because of their higher degree of freedom to move, in comparison with the one-dimensional case. One specific issue that is not yet satisfactory in the latter is that once an interaction occurs between a fully random agent and a deterministic one, the high probability that they will carry on interacting during the immediately subsequent iterations (since very likely they will remain close to each other) is an undesirable artifact. One idea to circumvent this problem is to allow some kind of shield to be built for a few iterations in the R-agent, just after its interaction with the D-agent, but the consequences of that have to be carefully evaluated before it becomes part of the model.

All in all, although DRIMA is operationally sound in its present conception, both as a model and as a system, there is certainly room for improvements and these are indeed under serious consideration. Additionally, an initial study was recently carried out to better understand a particular aspect of the system, namely, the speed with which each random agent had its movement pattern modified before becoming deterministic, as well as this convergence speed for the group of random agents as a whole [6]; however, a natural and essential step forward lies in the use of DRIMA to address specific questions involving the metaphors to which it refers. We hope that the qualitative appeal of the model that this article tried to convey might trigger such kinds of efforts.

■ Acknowledgments

DRIMA's materialization was catalyzed by conversations with one of my students, J. A. Poiani, who came to be the first intensive user of the system, for which I am very grateful; the notions of susceptibility and degree of influence that I mention in the article are his ideas, even though they have not (yet) been tried out in the system. I also express my gratitude to Wolfram Research, Inc., for having awarded me a *Mathematica* Academic Grant (No. 1149), without which DRIMA would still be just a non-materialized idea. And finally, I would like to acknowledge the financial support provided by FAPESP, Fundação de Amparo à Pesquisa do Estado de São Paulo (Proc. 2005/04696-3), and by Mack-Pesquisa, Fundo Mackenzie de Pesquisa (Edital 2007).

■ References

- [1] S. Wolfram, *The Mathematica Book*, 5th ed., Champaign, IL: Wolfram Media, 2003.
- [2] C. G. Langton, ed., *Artificial Life: An Overview*, Cambridge, MA: MIT Press, 1995.
- [3] G. Weiss, ed., *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, Cambridge, MA: MIT Press, 1999.
- [4] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, New York: Oxford University Press, 1999.
- [5] F. Varela, E. Thompson, and E. Rosch, *The Embodied Mind: Cognitive Science and Human Experience*, Cambridge, MA: MIT Press, 1991.
- [6] J. A. Poiani and P. P. B. de Oliveira, "Convergence of an Agent's Behaviour, through Its Interaction with Others, in a Multi-Agent Cellular System" (in Portuguese), in *Proceedings of the XXVII Iberian Latin American Congress in Computational Methods in Engineering (CILAMCE)*, CD-ROM, Universidade Federal do Pará: Belém-PA, Brazil, 2006.
- [7] R. Brooks, "Intelligence without Representation," *Artificial Intelligence*, **47**(1-3), 1991 pp. 139-159. doi:10.1016/0004-3702(91)90053-M.
- [8] M. Wooldridge, *Introduction to MultiAgent Systems*, New York: Wiley, 2002.
- [9] S. Wolfram, *A New Kind of Science*, Champaign, IL: Wolfram Media, 2002.
- [10] V. Dakshinamoorthy. "Coevolution of Competing Players in a Region." (Jan 21, 2010) www.wolframscience.com/summerschool/2005/participants/dakshinamoorthy.html.
- [11] A. Nair. "Mutating Turing Machines as a Metaphor for Interactive Behavior: Examining Coordination in Interorganizational Relationships." (Jan 21, 2010) www.wolframscience.com/summerschool/2005/participants/nair.html.
- [12] R. Zinkov. "Using Block Automata to Simulate Modular Robotics." (Jan 21, 2010) www.wolframscience.com/summerschool/2006/participants/zinkov.html.
- [13] T. Hogg, B. A. Huberman, and C. P. Williams, "Phase Transitions and the Search Problem," *Artificial Intelligence*, **81**(1-2), 1996 pp. 1-15.

P. P. B. de Oliveira, "DRIMA: A Minimal System for Probing the Dynamics of Change in a Reactive Multi-Agent Setting," *The Mathematica Journal*, 2010. [dx.doi.org/doi:10.3888/tmj.12-1](https://doi.org/10.3888/tmj.12-1).

About the Author

Pedro Paulo Balbi de Oliveira graduated as an Electronics Engineer from the Aeronautics Institute of Technology, Brazil (1981); he has an MSc in Applied Computing from the Brazilian Institute for Space Research (1986), and a DPhil in Cognitive Science from the University of Sussex, U.K. (1994). He has been a faculty member at Mackenzie Presbyterian University, São Paulo, Brazil, since 2001. His main research interests are cellular automata, applications of evolutionary computation, and artificial life.

Pedro P. B. de Oliveira

Universidade Presbiteriana Mackenzie

Faculdade de Computação e Informática & Pós-Graduação em Engenharia Elétrica

Rua da Consolação 896, Consolação

01302-907 São Paulo, SP—Brazil

pedrob@mackenzie.br