The Microsoft Research - University of Trento
Centre for Computational and Systems Biology

Technical Report CoSBi 25/2008

Graph Transformations and Game Theory: A Generative Mechanism for Network Formation

Matteo Cavaliere, Attila Csikász-Nagy, Ferenc Jordán
Microsoft Research - University of Trento
Centre for Computational and Systems Biology, CoSBi, Trento, Italy
{cavaliere, csikasz, jordan}@cosbi.eu

Abstract

Many systems can be described in terms of networks with characteristic structural properties. To better understand the formation and the dynamics of complex networks one can develop generative models. We propose here a generative model (named *dynamic spatial game*) that combines graph transformations and game theory. The idea is that a complex network is obtained by a sequence of node-based transformations determined by the interactions of nodes present in the network. We model the node-based transformations by using graph grammars and the interactions between the nodes by using game theory. We illustrate dynamic spatial games on a couple of examples: the role of cooperation in tissue formation and tumor development and the emergence of patterns during the formation of ecological networks.

1 Introduction

In several contexts, it has been shown that the structure of social, technological, biological and molecular networks share some non-random systems features [56], [48], [2]. An important issue is that networks typically change in time at both local and global level (typically by adaptation and transformation, respectively). Local changes make the emergence of global patterns possible, while global topology may constrain local dynamics, [49]. The coupling between local changes and global topologies of the networks is considered one of most important problem in the area, e.g., [2], [12]. In this paper we present a generative (rewriting) mechanism for studying the dynamic of complex networks (graphs), with the idea that networks are modified by means of node-based transformations that are “internal” to the networks (i.e., not due, for instance, to external perturbations). The approach proposed here is similar to what has been done in the area of natural languages where rewriting mechanisms (formal grammars) have been proposed to model the generation of complex sentences by the iterative applications of simple transformations (productions), e.g., [11], [42].

Specifically, in this paper, we propose a generative model based on graph grammars and game theory. The underlying idea is that complex networks are obtained by iterative node-based transformations determined by the interactions between the nodes of the networks. The node-based transformations are modeled by using graph grammars, while the interactions between the nodes are abstracted by using game theory.

In the proposed model, that we name *dynamic spatial game* (where ”spatial” may refer to both topographical and topological space), an initial network is iteratively modified in the following manner. Each node in the network has an associated *label (state)* that defines its type according to a specified game (e.g., whether the node is a cooperator node or a defector node). At each step of the dynamic spatial game, each node obtains a *value* that is calculated by considering the current state of the node and the states of other nodes in the network (e.g., the adjacent nodes). Then, based on the obtained value and on an *adopted nodal response*, each node executes a certain *node-based transformation* of the network.

A node-based transformation consists in replacing a node with new nodes (e.g., split a cooperator node into cooperator node and a defector node; split into two cooperator nodes; etc..) and embed the new nodes in the rest of the network in a certain specified manner (e.g., connect random, connect to everybody in the network, etc..).

The nodal responses and, consequently, the node-based transformations executed by each single node determine the transformations of the network and, ultimately, shape the overall structure of the network. Intuitively, the nodal responses adopted by the nodes can be seen as influenced by some internal property of the nodes (i.e. network- and context-independent attributes, like genetics).

The main advantage of the model proposed here is the fact that it is obtained by combining graph grammars and game theory, already well-studied tools. As we will see, this allows the cross-fertilization of tools and notions between different areas.

On the other hand, dynamic spatial games group ideas and generalize notions present in several fields. In fact, the model is essentially based on graph transformations and graph grammars, [40] where subgraphs can be replaced, in a recursive manner, by other subgraphs. There are approaches that explicitly take in account interactions of adjacent nodes (e.g., [57], [39], [46], [29]), but it is missing a graph grammar formalism where the interactions between the nodes in the graph are modeled by using game theory. This idea make our approach closer to the work done in spatial games, [35], and games on graphs, (e.g., [36], [45], [50]) that consider players of a game positioned on the nodes of a regular grid or on a graph. In these works, a game is played on a fixed regular grid or of a fixed graph. In our case, we have players positioned on the nodes of a network, but the underlying network is dynamically changed by the actions of the players (hence ”dynamic spatial game”). In this respect, our work is similar to network formation, [26], that investigates the importance of networks and their formation in economic situations. However, in network formation, the nodes have no “internal” state, while in our case, the nodes have a state that define their types (e.g., being a cooperator or defector) and the nodes can switch between different states. Moreover, in network formation the actions of the nodes is the reinforcing or removal of specific connections (see, e.g., [49], [44]) while in our case the transformations of the network are more general, since a node can be replaced by an arbitrary subgraph, embedded in some specific manner in the global network. This can lead to increase and decrease of the network size. Also in network formation (as used in the economic context) is usually assumed that a connection between the nodes need the consent (or contribution) of both nodes. In our case connections are established unilaterally by the nodes. In this respect there are similarities with the work in [19] but there the networks considered have a fixed dimension. There are also similarities between our approach and repeated games in the context of bounded rationality, [43]. In fact, in [43], a player can decide to switch between several states (e.g., from cooperator to defector), according to the obtained payoff, and this is modeled by using finite state automata. We generalize this concept by allowing a node to switch between several states, according to the obtained value, but also to be split in several nodes, and, possibly, to be deleted (this can be done, as in the proposed model, by using a generalization of finite state automata such as formal grammars, [25]). The relations between the proposed model and similar paradigms is illustrated in Figure 1.

As we can see from the previous discussions, the model is general and flexible enough to be applied to several areas and problems. In this paper we sketch two possible applications: one on tissue formation, studying the role of cooperation in cancer development, and another one on emerging patterns in ecological network development.

We first introduce the separate parts of the model: game theory in Section 1.1 and graph transformation (graph grammars) in Section 1.2. We then present the model in Section 2, present the above applications in Sections 3 and conclude in Section 4.

1.1 Game Theory

Game theory describes, in an abstract manner, the interactions between individuals (players) that use distinct strategies. The reward (payoff) a player gets depend on the chosen *strategy* and on the opponent's strategy. The strategic form of a game specifies the players of the game, their feasible actions (strategies) and the payoffs received by them for each possible combination of actions that could be chosen by the players.

A *2-players game* is formally specified, in the standard form, by giving a a 2×2 *payoff matrix*.

Moreover, for simplicity, we only consider *symmetric games*, [37], that are games where the identity of the players does not influence the obtained payoff. We also restrict the situation to 2-players games. In what follows, 2-players symmetric games are simply referred as games. A game G is composed by a finite set of strategies S and a payoff function $E : S \times S \to \mathbb{R}$. Precisely, $E(i, j)$ denotes the payoff obtained by a player adopting strategy $i \in S$ against an opponent player adopting strategy $j \in S$. When it is not clear from the context, for a game G, S is denoted by S_G and $E(i, j)$ is denoted by $E_G(i, j)$.

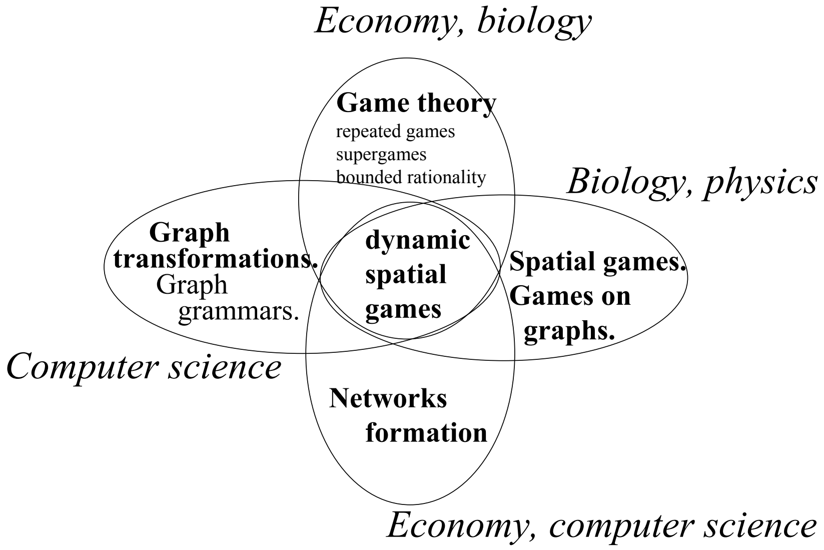


Figure 1: The relations between dynamic spatial games and other similar paradigms.

	cooperate (C)	*defect (D)*
cooperate (C)	*4*	*1*
defect (D)	*5*	*2*

Figure 2: Payoff matrix of the Prisoner's dilemma. The matrix is read in the following manner. A player that defect (defector) against a player that cooperates (cooperator) gets 5 as payoff, $E(D, C) = 5$, while the cooperator gets 1, $E(C, D) = 1$. If both players are defector then both get 2, $E(D, D) = 2$; if both players are cooperators, then both gets 4, $E(C, C) = 4$. The set of strategies of the game is $S = \{C, D\}$.

An introduction to game theory can be found in [37]. In Figure 2 is reported the matrix form of the prisoner's dilemma game.

1.2 Graph Transformation

The field of *graph transformation* consists in the study of the formalisms that describe the dynamics of graphical structures (graphs, visual sentences, etc..). It is a very active research area with applications to concurrent systems, databases, programming languages, and biology (e.g., see the proceedings of the latest international conference on graph transformation [16] and the webpage [58] for an overview of the current applications in the area).

In this paper the terms "graph" and "network" indicate a finite undirected node labeled graph. A *graph* is formally given in the form $g = (V, E, \Sigma, lab)$ where V is a finite set of nodes, $E \subseteq \{\{x, y\} \mid x, y \in V\}$ is a set of undirected *edges*, Σ is a (finite nonempty) alphabet of *node labels*, and $lab : V \to \Sigma$ is a *labeling function*; for $v \in V$, $lab(v)$ is the label of node v. All edges in the set $\{\{x, y\} \in E \mid x \in V\}$ are called *adjacent* to x (these are the edges that have as endpoint node x). All nodes $adj(x) = \{y \in V \mid \{x, y\} \in E\}$ are called *adjacent nodes* to x. When it is not clear from the context, we denote the components of a graph g by V_g, E_g, Σ_g and lab_g.

For an alphabet Σ, the class of all graphs with the alphabet of node labels equal Σ and i nodes is denoted by $g_{i,\Sigma}$. The *class of all graphs with the alphabet of node labels equal* Σ *is then denoted by* g_Σ. Formally, we have that $g_\Sigma = g_{0,\Sigma} \cup g_{1,\Sigma} \cup \cdots \cup g_{i,\Sigma} \cup \ldots$ where $g_{0,\Sigma}$ is the empty graph, having no nodes and no edges.

Among the general models of graph transformation, one of the most studied, with several characterizations is called *graph grammar* (the reader can find a complete coverage of the area in the handbook [40]). The basic idea of graph grammars is to extend to graphs the theory of formal languages, formal grammars and automata theory, [25], used in the context of natural languages and theory of computation. Graph grammars were first introduced in [38] to solve picture processing problems.

The basic unit of a graph grammar is a *production.*

The role of a production is to replace one subgraph by another. In this way, a graph g is transformed in a new graph g'. A production is applied by replacing a subgraph m of g by a graph d that is then embedded into the remainder of g, i.e., the graph that remains after removing m from g. This process depends on the specification of the embedding, that is how the graph d is inserted in the remainder of g. We say that g is the *host graph* (where the production is applied), m is the *mother graph* (the removed graph) and d is the *daughter graph* (the inserted graph). The removal of graph m from g includes the removal of all edges of g that are adjacent to nodes of m. The *embedding function* specifies how to embed d into the remainder of g. In this paper we consider the special case of *node-replacement productions*, simply called *productions*, where the mother graphs consist of *one node only.*

We give now a more formal definition of the productions (we follow the definition and the terminology used in the area, see, e.g., [41]).

Let Σ be a set (alphabet) of node-labels. A *production over* Σ is a triple $\pi = (a, \beta, \gamma)$ with $a \in \Sigma$ is a node-label and is called the *left-hand side* of π; β is a labeled graph and is called the *right-hand side* of π (in particular, β can be the empty graph). Often π is written in the form $a \rightarrow \beta$. The symbol $\rightarrow$ is read as "is rewritten as". The component γ is the *embedding* function that specifies the *embedding* used by the production.

The rewriting (replacement) of a node n with label a (*mother node*) in a host graph g by using the production $\pi = (a, \beta, \gamma)$ is done in the following way.

(1) The node n is removed from g, together with all edges adjacent to n, leaving the *rest graph* RG.

(2) A subset EA of nodes of RG is selected based on the function γ. This subset of nodes is called *embedding area.*

(3) A graph $\overline{\beta}$ isomorphic to the *daughter graph* β is added to RG ($\overline{\beta}$ replaces node n). [1]

(4) Using the embedding function γ the graph $\overline{\beta}$ is embedded in RG, i.e., edges are established between some nodes in $\overline{\beta}$ and some nodes of EA. [2]

Examples of productions are given in Figure 3. An application of a production is presented in Figure 4.

Given a set Γ of productions over Σ, we denote by Γ_a the set of productions in Γ having as left-hand side the label a. The set of all left-hand sides of the productions in Γ is denoted by $left(\Gamma)$.

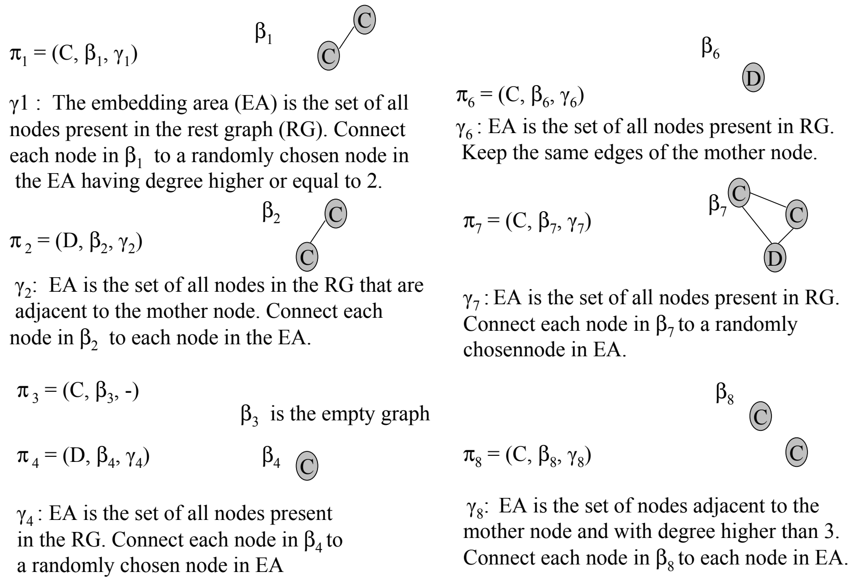
2 Dynamic Spatial Games

As discussed in the Introduction, we propose a model where individual nodes "act" based on adopted nodal responses and on their interactions with other nodes present in the network. An (infinite) iteration of such "actions" is then referred as dynamic spatial game.

A *dynamic spatial game* is started from an initial network composed by

[1] We suppose that nodes in $\overline{\beta}$ are disjoint from those in RG.

[2] The various types of productions differ mainly in the way the embedding function is defined. The definition of such function is crucial in the description of the productions and the final theory. There are many types of productions in literature where several embedding functions have been defined and investigated. For instance, one can say that the embedding area EA consists only in the nodes that are adjacent to the removed node and only edges between the nodes in the daughter graph and nodes in EA can be added (this class of productions is well-studied and is generally referred as context-free, e.g., [18]). We do not enter here in details. The reader can find a survey on the classes of productions and embedding functions in the handbook of graph grammars [40] and in the survey [7].



$\pi_1 = (C, \beta_1, \gamma_1)$ β_1

$\gamma 1$: The embedding area (EA) is the set of all nodes present in the rest graph (RG). Connect each node in β_1 to a randomly chosen node in the EA having degree higher or equal to 2.

$\pi_2 = (D, \beta_2, \gamma_2)$ β_2

γ_2: EA is the set of all nodes in the RG that are adjacent to the mother node. Connect each node in β_2 to each node in the EA.

$\pi_3 = (C, \beta_3, -)$

β_3 is the empty graph

$\pi_4 = (D, \beta_4, \gamma_4)$ β_4

γ_4 : EA is the set of all nodes present in the RG. Connect each node in β_4 to a randomly chosen node in EA

$\pi_6 = (C, \beta_6, \gamma_6)$ β_6

γ_6 : EA is the set of all nodes present in RG. Keep the same edges of the mother node.

$\pi_7 = (C, \beta_7, \gamma_7)$ β_7

γ_7 : EA is the set of all nodes present in RG. Connect each node in β_7 to a randomly chosennode in EA.

$\pi_8 = (C, \beta_8, \gamma_8)$ β_8

γ_8: EA is the set of nodes adjacent to the mother node and with degree higher than 3. Connect each node in β_8 to each node in EA.

Figure 3: We illustrate some productions over the alphabet $\{C, D\}$. Each production is described by a triple. The first component denotes the label of the node to replace. The second component denotes the graph that replaces the node. The third component, given in a narrative way, denotes the embedding conditions that specify how the inserted graph is inserted in the network. As one can see, in general, the labels of the nodes in the right side may be different from the one in the left side. Application of production π_1 is described in Figure 4.

an arbitrary number of nodes. Each node has an associated label (state) that denotes its type (strategy) according to a specified game (e.g., the node is a cooperator or a defector).

At each step of the dynamic spatial game each node in the network receives a value that can generally depend on its current state, on the states of other nodes in the network, and on the network topology. Based on the obtained values, and on the adopted nodal responses, each node executes an "action" that consists in: (*i*) being replaced by zero, one or more new nodes (e.g., a cooperator is replaced by two defectors); (*ii*) connect the new nodes to the rest of the network, in some specified manner (e.g., in a random manner).

Summarizing, the key aspects of a dynamic spatial game are then the following ones:

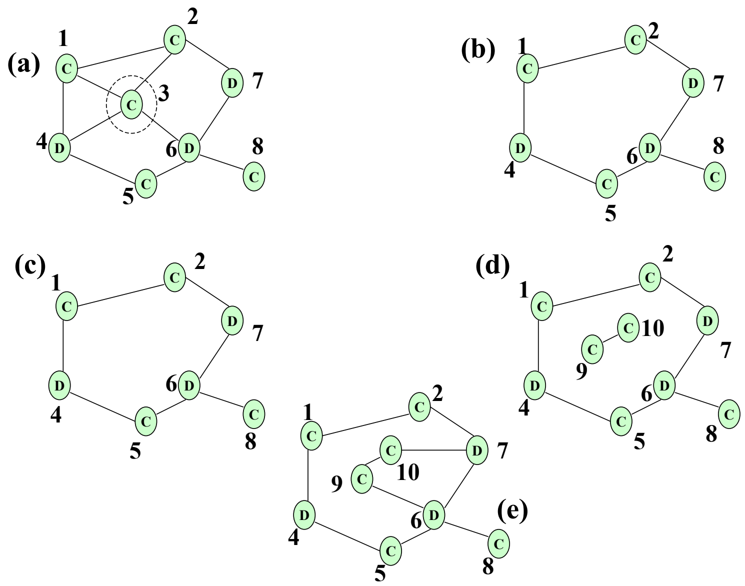


Figure 4: We show in detail the replacement of a node by using the production π_1 presented in Figure 3. The node to be replaced is selected and removed, together with the adjacent edges (steps (a) and (b)). The rest graph (RG) obtained is shown in (b). At step (c) the embedding area (EA) is shown, selected as specified in the embedding function γ_1. In this case, EA consists of all nodes present in RG. At step (d) a graph isomorphic to β_1 is added to RG. At step (e) edges between the inserted graph and nodes of EA are added, as specified in γ_1. In this case, the embedding function γ_1 specifies that one edge must be established between each node of the inserted graph and a randomly chosen node in the EA with degree higher or equal to 2. In the Figure, node 9 is connected to node 6, while node 10 is connected to node 7. The new graph obtained is (e).

- The dynamic spatial game is executed on a network.
- The nodes in the network have associated a label (state) that denotes the strategy of a game.
- Each node obtains a value that can depend on its own state, on the states of other nodes in the network and on the network topology.
- Each node executes an action according to the obtained value and the adopted nodal response (the nodal response is an algorithm that specifies an action for each value that the node can get). The action consists in the replacement of the node by means of an appropriate production.

- Nodal responses are passed to successive generations.

Technically, a dynamic spatial game is obtained by composing a set of productions, a game and a value function.

Formally, a *dynamic spatial game* (in short, *DSG*) is the construct $\mathcal{B} = (G, \mathcal{F}, \mathcal{R})$ where:

- G is a 2-player symmetric game with set of strategies S.

- $\mathcal{F}$ is a family $\{f_g \mid g \in g_S\}$ of *value functions* with $f_g : V_g \to \mathbb{R}$. [3]

- $\mathcal{R}$ a finite set of pairs (Γ, η) with: Γ is a set of productions over S; $\eta = \{\eta^x_a \mid a \in S, x \in \mathbb{R}\}$, where each η^x_a is a probability distribution function over Γ_a. [4] Each pair in $\mathcal{R}$ is called *nodal response.*

A DSG starts from an *initial network* $g^0 \in g_S$. The mapping $\mu^g : V_g \to \mathcal{R}$, for $g \in g_S$, associates to each node of g a nodal response from $\mathcal{R}$ (the mapping μ^{g_0} is called *initial mapping*).

A DSG consists in a sequence of *transition steps* (simply called steps). Each step transforms the *current network* $g \in g_S$ into a new network $g' \in g_S$. A step is executed in the following manner. A node n is selected in V_g (suppose that $lab_g(n) = a$ and that $\mu^g(n) = (\Gamma, \eta)$). The node n is then replaced by using the production $\pi \in \Gamma_a$ with probability $\eta^x_a(\pi)$, where $x = f_g(n)$. After the replacement, the new graph g' is obtained. The nodes $n_1, \cdots, n_k$ that have replaced node n are still associated to the strategy (Γ, η), i.e., $\mu^{g'}(n_1) = \cdots = \mu^{g'}(n_k) = \mu^g(n) = (\Gamma, \eta)$. [5]

An example of a DSG is given in Figure 5. Several steps of the DSG are described in Figure 6.

[3] A value function is used to associate a value to each node of a graph taken in g_S, i.e., the class of graphs with the nodes labeled by the strategies of G. Therefore, $f_g(n)$, with $g \in g_S$, is the value associated in the graph g to the node n. One can define arbitrary value functions depending on the application. In this paper we consider a value function as used in spatial games [35]: The value of a of node n in a graph g is calculated as the sum of the payoffs resulting from playing with all adjacent nodes in the graph g. In this case, for any any $g \in g_S$ and any $n \in V_g$ we have that $f_g(n) = \sum_{j \in adj_g(n)} E_G(lab_g(n), lab_g(j))$. Because of its semantics, we call such function *node-additive.*

[4] The function η^x_a specifies the probability that a node labeled by a and with value x is replaced by using a certain production. Specifically $\eta^x_a(\pi)$ is the probability of replacing a node with label a and value $x \in \mathbb{R}$ by means of the production $\pi \in \Gamma_a$. The definition is given in a very general way to allow many possible responses (e.g., best-response, [26]) and to include restrictions (e.g., costs).

[5] We have defined a step of a dynamic spatial game. A sequence of steps constitutes a dynamic spatial game. In this case one has to specify a way to select the nodes in the

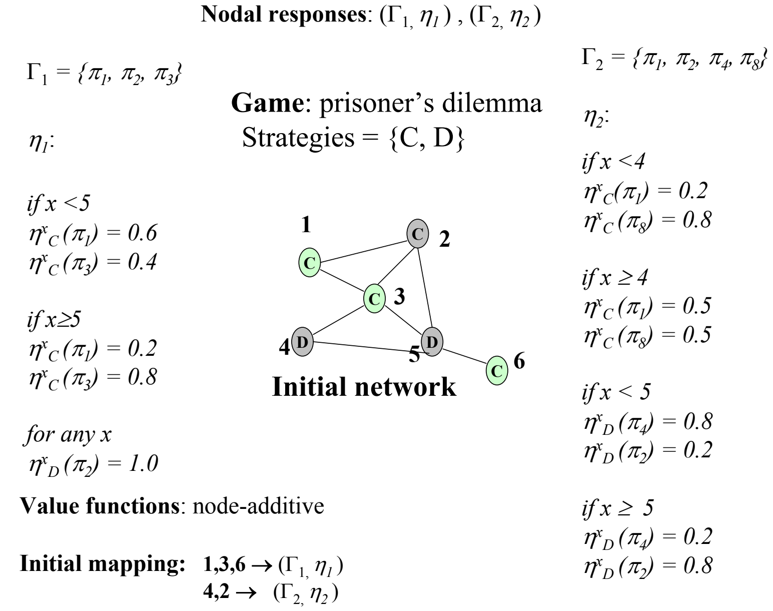


Figure 5: We present an example of a dynamic spatial game $\mathcal{B}$. The figure shows its components. The game G is the prisoners' dilemma with set of strategies C, D and payoffs matrix defined as in Figure 2. The value function is node-additive. There are two nodal responses: the pair (Γ_1, η_1) and the pair (Γ_2, η_2) (the productions in Γ_1 and Γ_2 are described in Figure 3). The initial mapping associates to each node of the initial network a certain nodal response. In this case nodes $1, 3$ and 6 use the nodal response (Γ_1, η_1), while the rest use (Γ_2, η_2). Some steps of $\mathcal{B}$ are presented in Figure 6.

3 Old and New Questions

DSGs are obtained by combining two well-known areas: graph grammars and game theory. This has two advantages: it extends the notion of graph grammars and allows the cross-fertilization of questions and tools between different areas. Here, we want to briefly discuss some of the possibilities.

A major problem in graph grammars concerns the mathematical characterization of the family of graphs generated by a certain class of grammars (where generated means obtained by iterations of the productions). Following the natural language terminology, these families of graphs are called the

current network, at each step (all, some of them, etc..). For instance, one can consider sequences of *parallel* steps where each node in the current network must be considered before considering the new introduced ones. Other "update dynamics" can be considered (e.g., as done in cellular automata). We do not enter in details, but, generally, the ways nodes are considered in the current network can have an important effect on the obtained networks.

"languages" generated by the graph grammars. In this respect the literature is very rich, e.g.,[40]. Many grammars have been investigated and several mathematical characterizations for the generated languages have been produced. In several cases one can, for instance, predict if a certain graph can be generated by a certain graph grammar (see, [7]), even in an efficient manner (see, e.g., [8], [31]). These results could be re-used in DSGs based on the fact that nodal responses are essentially graph grammars productions. In this way, one could characterize the networks obtained when all nodes use the same nodal response. We expect that extending the results obtained for graph grammars, one could predict, for restricted cases, the type of networks obtained when all nodes use the same response. Most probably, for this purpose, one has to restrict the core model by considering specific nodal responses (e.g., productions with particular embedding functions) and specific value functions. In this respect, there are several interesting candidates such as context-free graph grammars, with well-known mathematical characterizations (see, e.g., [10], [18]). Also the strong theoretical foundations of graph grammars can have significant advantages in a formal study of the constructed models. For instance, one can investigate confluence – the ability to reach, independently of the initial network and of the update dynamics, a certain specified range of networks. Which are the nodal responses that, when adopted by all nodes, guarantee confluence? The topic of confluence has been widely studied in the theory of graph grammars in several applications, see, e.g., [10], [23]. Another important question concern the robustness of a certain nodal response, that is the ability for a network to survive to perturbations, that in our case means to add or remove nodes using different nodal responses. Again, a similar question has been approached in the area of graph grammars in a very different context: the integrity of databases and of software architectures, see, e.g., [15], [33], [24] and [14].

One of the most important line of research in network formation is to understand how players can maximize their cost-benefit ratio in building a network, and at the same time to have a network that satisfies certain properties (see, e.g., [51]). These types of problems could be investigated in DSGs in a more general form. In fact, in DSGs, a node can, not only, change its connectivity, but also splits in several nodes, each of them with its own internal state. These features would probably require an extensions of the concepts introduced in the area of network formation, in particular of Nash equilibrium and of the price of anarchy, that have been defined for networks with fixed number of nodes (see, e.g, [51]). Moreover, using the proposed model one can also represent the situations of "global" games, where external "metaplayers" "own" several nodes in the network. Such metaplayers can control the splittings of the nodes, their removal, their connectivity and their internal states with the final goal, as in network formation, to obtain certain types of networks at certain costs (that can include the costs of splitting, changing states, reconnect, the number of owned nodes, etc..). For instance, in Figure 6, one can imagine two metaplayers, owning the nodes in the initial network. The first one owns the nodes $1, 3$ and 6 and the second one owns the rest. Each metaplayer "manages" the owned nodes by the assigned nodal responses.

Notice also that well-known embedding mechanisms (e.g., preferential attachment [6]) could be used in nodal responses. A natural question is whether these nodal responses are actually evolutionary stable strategies, [32], in the sense that are resistant to small mutant invasion (e.g., nodes using different nodal responses). This would lead to an evolutionary explanation of why such nodal responses can actually spread and survive, and, consequently, why certain types of networks are more frequent than others.

On the other hand, the used notion of production, stresses the fact that nodal responses are composed by two components that define (*i*) when to split a node and/or change its state and (*ii*) when to change the connectivity of a node. This can pose the following insights: There could exist a trade-off between these two components. Such trade-off could lead to cases when, depending on the actual conditions, one of these components dominates. There are examples in nature that seem to support the existence of such trade-off. For instance, brain neurons dynamically change their interactions (connectivity) but rarely divide [21], or butterfly species seem to be either sedentary and good competitors or, alternatively, mobile and weak competitors [54].

Despite its simplicity the presented model is enough flexible to be applicable in several areas. To illustrate this we sketch two possible applications, concerning tissue formation (cell-to-cell communication) and the development of patterns in ecological networks. We avoid a formal definition of the models, but rather we prefer to stress the intuitive explanation of the proposed applications.

3.1 Emerging patterns in ecological networks

Ecological systems are composed of several components (e.g., species) and several types of interspecific interactions among them. These include various combinations of signed effects between two species that can, e.g., act "positively" or "negatively" between them. For instance, we can classify the interactions according to the effects of the involved species: (+/+: mutualism, -/-: competition, +/-: predator-prey or host-parasite interaction, +/0: facilitation, -/0: amensalism). It is a classical problem in ecology how to

[6] Preferential attachment, defined in [4], is a generative mechanism that allow the creation of scale-free networks.

study simultaneously these multiple interactions (see [55]). For example, it is not easy to compare the relative effects of a pollinator and an herbivore on a particular plant species. Thus, it is a great challenge to model signed ecological networks.

Moreover, the paleontological record suggests some general patterns for the evolution of ecological interactions, [53]. For example, a high proportion of mutualistic interactions seem to have evolved from higher-order interactions, or pollination is typically a result of herbivory. Coexisting species co-evolve in a community context: the transformation (change) of interactions modifies the network and the network constraints local dynamics. An important question is how local or how global are these kinds of transformations, [28].

In this section we sketch an example that shows how both aspects – species interactions and transformations of interactions – could be modelled using DSGs.

The basic idea is to consider a food-web as a network, with nodes representing species and the edges that represent species and interspecific interactions. We associate to each node a label, $+$ or $-$, that indicates if the species acts positively or negatively, respectively, on its partners (in short, a species labelled by $+$ is called *positive species*, a species labelled by $-$ is called *negative species*; this simplification, that is open for further refinement, aims to make a difference between species that tend to be mutualists and species that tend to be antagonistic partners).

We allow the possibilities of structural changes of the food-web by considering speciation (creation of a new species), extinction and change of interactions partners. These structural changes in the food-web can be implemented by opportune productions that determine the local dynamics of the food-web and that are discussed in Figure 7.

Partially inspired by [54], we suppose that the species are grouped in two main classes with distinct ways of changing their partners: *specialists* and *generalists*. In a narrative way, one can say that specialists (*i*) build only a few links to others, (*ii*) prefer local partners and (*iii*) behave in a conservative way (do not change interactions easily); generalists (*i*) prefer to have more connections, (*ii*) explore a larger area (in our model, mobility is replaced by topological distance) and (*iii*) change more easily partners. We can also assume that there are some genetical tendency of every species to evolve in a more specialist or more generalist way: in this way specialists and generalists can be modelled as nodes that use distinct nodal responses, in an appropriate DSG. A DSG with specialists and generalists is illustrated in Figure 8.

The appropriate tuning of the payoffs matrix of the nodal responses in Figure 8 can capture several interesting situations leading to important questions in ecology. In what follows we shortly enumerate some of them:

(1) how results depend on the actual values in the payoffs matrix, (2) which qualitative changes of the payoffs matrix provide confluency, (3) under which conditions (productions and payoffs) will the ”tendency towards structural balance” [22] emerge (if any), (4) how does the outcome of the DSG depends on the initial positions of specialists and generalist nodes, (5) how do the two subgraphs composed of generalists and specialists change in time, (6) under which conditions do ”broker” nodes appear in the network (e.g., nodes with only a few links but very strong indirect neighbourhood), (7) how do the productions (e.g., embedding functions) affect emerging food web topologies (e.g., scale-free networks, [2]), (8) which is the relevant size of neighborhoods responsible for local dynamics (i.e., how “local is the local”), (9) under which conditions specialists and generalists nodes are more successful (e.g., in terms of their ratio within the network), (10) which conditions make it easier to reach global optima on a rugged landscape (i.e., providing shallow valleys), and, as a general question, (11) how contingent is global topology with local mechanistic rules. As a particular question, (12) we are still interested in the topological position of specialists and generalists nodes in a ”mature” network: are they aggregated, is any of them more central, and what kind of subgraphs they form?

Finally, notice that the use of productions allows to consider food-webs with a dynamically changing number of species and of interactions; this is rather different from other approaches in the area that have considered the degree of the nodes as constant, see, e.g. [30].

3.2 Games and Developmental Biology

A living cell can rest in G0 state, proliferate or die, depending on the external and internal signals it receives. Tumor formation is a long process, where cells collect multiple mutations to get fully independent of the environment and proliferate without control [20]. In [3] Axelrod and colleagues proposed the hypothesis that cells with a few early stage mutations, cooperate with each other to help further steps of malignant tumor formation. Cooperation among partially mutated cells may take the form of by-product mutualism by sharing resources and growth-factors or anti-apoptotic agents. The important point of this hypothesis is that at the early steps of tumor formation individual cells do not need to accumulate all of the mutations to be independent of external signals. Cells that have accumulated only a few relevant mutations may cooperate with other partially mutated cells to form a community of cells that can survive, increasing the chance to get further mutations and independence from the surrounding tissue.

We show how one can formalize and investigate the hypothesis proposed in [3] by using an appropriate DSG.

Following the approach proposed in [5] bidimensional cellular tissues can be modeled by using triangulated graphs. In this case the nodes are interpreted as cells. Cell division and cell death can be seen as operations on triangulated graphs and mutations in the cell line can be also simulated by the appropriate productions, as illustrated in Figure 9.

We can couple the productions described in Figure 9 with an appropriate game and consider the labels of the cells (nodes) as strategies of the game. For instance, the strategies of the game could represent the possible states of a cell. As a simplification, we only consider that cells can be in *normal state*, in a *partially mutated state* and in an *independent mutated state.* The game can be then used to abstract the interactions between these types of cells. In this way, the cooperation between cells in different states (e.g., partially mutated cells) can be then tuned in the payoffs matrix of the game. The corresponding DSG is discussed in Figure 10.

The DSG presented in Figure 10 can be used to formalize more general questions in developmental biology and tissue formation. In fact, a natural question concerns the relations between the game and the properties of the tissue (triangulated graph) obtained. Linking then to morphogenesis is rather direct as it has been showed the importance of competition between cells in drosophila wing formation, [47], as well in higher eukaryotes, [1].

4 Conclusions

Although we know more and more about the macroscopic topological properties of static networks, we need to much better understand network transformations in terms of simple local dynamics and emergent, global constraints: This problem is highly relevant in a variety of issues ranging from tumor formation to food web assembly, as illustrated in Section 3.1 and Section 3.2 and from the origin of social insect colonies [6] to homeland security [27]. The introduced model seems to be a general, theory-based framework for studying the mechanistic background of local (node-based) network transformations. The combination of graph grammars and game theory is original and we expect that much theoretical work can be done to import the results obtained in the area of graph grammars to this new framework. This would enrich the theory of graph transformations with interesting multidisciplinary questions and, as discussed in Section 3, would provide efficient computational tools and algorithms to verify and predict the generation of certain types of networks, for given classes of DSGs. Moreover, graph grammars can be efficiently implemented on computers given their "algorithmic/automata-based" nature (see, e.g., the numerous tools in [58]) and, in this respect, a software that can run several DSGs is currently in development. On the other hand, several extensions of the introduced model are possible. One can apply the same idea and using different kinds of graph grammars, e.g., that allow the replacement of entire subgraphs, [40] – this would allow to model, for instance, fusion of nodes. One could also consider weighted networks (e.g., by adding or removing weights to a particular edge) and use directed edges by allowing directed interactions (in fact, graph grammars have also been defined and investigated with directed edges, [40]). One could also consider particular classes of productions by using embedding functions that model the "semi-global" approach proposed by Harary [22], where graph dynamics is ruled by loop sign, i.e., determined neither locally nor globally, or to consider endogenous link formation as done in the area of computational economics, [52].

Acknowledgments The authors would like to thank Ivan Mura and Tarcisio Fedrizzi for very helpful discussions.

References

[1] J.M. Abrams, M.A. White, Coordination of cell death and the cell cycle: linking proliferation to death through private and communal couplers, *Current opinion in cell biology*, 16, 6, 2004.

[2] R. Albert, A.-L. Barabási, Statistical methods of complex networks, *Rev. Mod. Phys.*, 74, 2002.

[3] R. Axelrod, D.E. Axelrod, K.J. Pienta, Evolution of cooperation among tumor cells, *PNAS*, 103, 36, 2006.

[4] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, *Science*, 286, 1999.

[5] S. Bar-Duvdevani, L. Segel, On topological simulations in developmental biology, *Journal of Theoretical Biology*, 166, 1, 1994.

[6] A. Bhadra, F. Jordán, A. Sumana, S. Deshpande, R. Gadagkar, A comparative social network analysis of wasp colonies and classrooms: linking network structure to functioning, *Ecological Complexity*, in press.

[7] D. Blostein, H. Fahmy, A. Grbavec, Issues in the practical use of graph rewriting, *Proc. int. workshop on graph grammars and their application to computer science*, LNCS 1073, 1994.

[8] F.J. Brandenburg, On polynomial time graph grammars, *STACS 88*, LNCS 294, 1988.

[9] H. Bunke, Programmed graph grammars, *Proc. int. workshop on graph-grammars and their application to computer science and biology*, LNCS 73, 1978.

[10] B. Courcelle, An axiomatic definition of context-free rewriting and its application to NLC graph grammars, *STACS*, LNCS 294, 1988.

[11] N. Chomsky, *Aspects of the theory of syntax*, Cambridge, MA, MIT Press, 1965.

[12] S.N. Dorogovtsev, J.F.F. Mendes, *Evolution of Networks - From biological nets to the internet and WWW*, Oxford University Press, 2003.

[13] H. Ehrig, G. Engels, H.-J. Kreowski, G. Rozenberg Eds., *Handbook of graph grammars and computing by graph Transformation, Vol.2: Applications, Languages and Tools*, World Scientific, 1999.

[14] H. Ehrig, H.-J. Kreowski, U. Montanari, G. Rozenberg Eds., *Handbook of graph grammars and computing by Graph Transformation, Vol.3: Concurrency, Parallelism and Distribution*, World Scientific, 1999.

[15] H. Ehrig, H. Kreowski, Applications of graph grammars theory to consistency, synchronization and scheduling in database systems, *Information Systems*, 5, 1980.

[16] H. Ehrig, R. Heckel, G. Rozenberg, G. Taentzer Eds., *Proc. Graph transformation, ICGT2008*, LNCS 5214, 2008.

[17] H. Ehrig, G. Taentzer, Computing by graph transformation: A survey and annotated bibliography, *Bullettin of the EATCS*, 59, 1996.

[18] J. Engelfriet, Context-free graph grammars, in [42].

[19] A. Fabrikant, A. Luthra, E. Maneva, C. Papadimitriou, C. Shenker, On a network creation game, *Proc. ACM Symp. of distributed systems*, 2003.

[20] D. Hanahan, R. Weinberg, The hallmarks of cancer, *Cell*, 100, 1, 2000.

[21] G.I. Hatton, Function-related plasticity in hypothalamus, *Annual review of neuroscience*, 20, 1997.

[22] F. Harary, A structural analysis of the situation in the Middle East in 1956, *The Journal of Conflict Resolution*, 5, 1961.

[23] R. Heckel, J.M. Kü ster, G. Taentzer, Confluence of typed attributed graph transformation systems, *Graph transformation*, LNCS 2505, 2002.

[24] D. Hirsch, P. Inverardi, U. Montanari, Graph grammars and constraint solving for software architecture styles, *Proceedings on the third international workshop on software architecture*, ACM Press, 1998.

[25] J.E. Hopcroft, R. Motwani, J.D. Ullman, *Introduction to automata theory, languages and computation*, Addison-Wesley, 2001.

[26] M.O. Jackson, A survey of models of network formation: Stability and efficiency, in *Group formation in economics: networks, clubs, and coalitions*, Cambridge University Press, 2005.

[27] F. Jordán, Network analysis: linking parts to the whole in nature and society, in R. Sagarin, T. Taylor Eds., em Darwinian security - Perspectives from ecology and evolution, University of California Press, Berkeley and Los Angeles, 2008.

[28] F. Jordán, I. Scheuring, Searching for keystones in ecological networks, *Oikos*, 99, 2002.

[29] E. Klavins, Graph grammars for self assembling robotic systems, *IEEE Int. conference on robotics and automation, ICRA04*, 2004.

[30] Á. Kun, I. Scheuring, Evolution of cooperation on dynamical graphs, *Biosystems*, in press.

[31] T. Lengauer, E. Wanke, Efficient analysis of graph properties on context-free graph languages, *15th Conference on automata, languages and programming*, LNCS 317, 1988.

[32] J. Maynard Smith, *Evolution and the theory of games*, Cambridge University Press, 1982.

[33] D. Le Metayer, Describing software architecture styles using graph grammars, *IEEE Transactions on software engineering*, 24, 7, 1998.

[34] N. Nisan, T. Roughgarden, E. Tardos, V.V. Vazirani, *Algorithmic game theory*, Cambridge University Press, 2007.

[35] M. Nowak, R.M. May, Evolutionary games and spatial chaos, *Nature*, 359, 1992.

[36] M.A. Nowak, *Evolutionary dynamics*, Harvard University Press, 2006.

[37] M.J. Osborne, *An introduction to game theory*, Oxford University Press, 2003.

[38] J. Pfaltz, A. Rosenfeld, Web grammars, *Proc. Joint conference on artificial intelligence*, 1969.

[39] J. Rekers, A. Schurr, Defining and parsing visual languages with layered graph grammars, *Journal of visual languages and computing*, 8, 1, 1997.

[40] G. Rozenberg Ed., *Handbook of graph grammars and computing by graph transformation, Vol.1: Foundations*, World Scientific, 1997.

[41] G. Rozenberg, An Introduction to NLC way of rewriting graphs, in *Graph grammars and their application to computer science*, LNCS 291, Springer, 1987.

[42] G. Rozenberg, A. Salomaa Eds., *Handbook of formal languages, Vols 1-3*, Springer, 1997.

[43] A. Rubinstein, Finite automata play the repeated prisoner's dilemma, *Journal of Economic Theory*, 21, 1986.

[44] F.C. Santos, J.M. Pacheco, T. Lenaerts, Cooperation prevails when individuals adjust their social ties, *PLoS Computational Biology*, 2, 10, 2006.

[45] F.C. Santos, J.M. Pacheco, T. Lenaerts, Evolutionary dynamics of social dilemmas in structured heterogenous populations, *PNAS*, 9, 193, 2006.

[46] H. Sayama, Generative network automata: A generalized framework for modeling complex dynamical systems with autonomously varying topologies, *Proceedings of the 2007 IEEE Symposium on Artificial Life*, 2007.

[47] P. Simpson, G. Morata, Differential mitotic rates and patterns of growth in compartments in the Drosophila wing, *Dev. Biol.*, 85, 2, 1981.

[48] S.H. Strogatz, Exploring complex networks, *Nature*, 410, 2001.

[49] B. Skyrms, R. Pemantle, A dynamic model of social network formation, *PNAS*, 97, 16, 2000.

[50] G. Szabo, G. Fáth, Evolutionary games on graphs, *Physics reports*, 446, 4-6, 2007.

[51] E. Tardos, T. Wexler, Network formation games and the potential function method, in [34].

[52] L. Tesfatsion, Structure, behaviour, and market power in an evolutionary labor market with adaptive search, *Journal of Economic Dynamics and Control*, 25, 2001.

[53] J.N. Thompson, *Interaction and coevolution*, Wiley and Sons, New York, 1982.

[54] C. Thomas, Dispersal and extinction in fragmented landscapes, *Proceedings of the Royal Society*, London, series B, 267, 2000.

[55] V. Vasas, F. Jordán, Topological keystone species in ecological interaction networks: considering link quality and non-trophic effects, *Ecological Modelling*, 196, 2006.

[56] D.J. Watts, S.H. Strogatz, Collective dynamics of small-world networks, *Nature*, 393, 1998.

[57] D-Q. Zhang, K. Zhang, J. Cao, A context-sensitive graph grammar formalism for the specification of visual languages, *The Computer Journal*, 2001, 44, 3, 2001.

[58] www.gratra.org

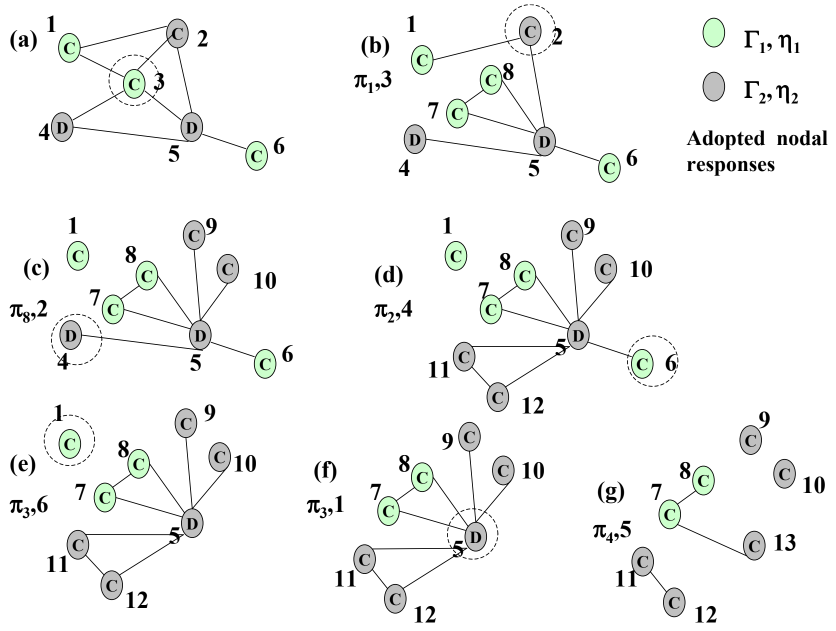


Figure 6: We present a (parallel) step of the DSG $\mathcal{B}$ defined in Figure 5, starting from the initial network (a). In a parallel step, all the 6 nodes present in the initial network must act according to the adopted nodal responses. Nodes are selected in an arbitrary order. A node gets a value calculated using the value function specified in $\mathcal{B}$. The node is then replaced by using the production specified in the adopted nodal response (in the Figure it is indicate, at each step, the selected node and the applied production). For instance, the transition from network (a) to network (b) is obtained in the following way. Node 3 is selected. The value of node 3 is calculated as $E_G(C,C)+E_G(C,C)+E_G(C,D)+E_G(C,D) = 10$ (i.e., value function of $\mathcal{B}$ is node-additive). Node 3 uses the nodal response (Γ_1, η_1) as specified in $\mathcal{B}$. Based on the obtained value 10 and on the adopted response (Γ_1, η_1) node 3 is replaced by using production π_1 with probability 0.2 (in fact, see Figure 5, for η_1 we have that $\eta_C^{10}(\pi_1) = 0.2$). The inserted nodes 7 and 8 use the same nodal response of the replaced mother node 3 (we suppose that nodal responses are genetically coded and transferred to successive generations). We can also look at the transition from network (c) to (d). In this case node 4 is selected. The value of the node is 1. The node uses the nodal response (Γ_2, η_2) as specified in $\mathcal{B}$. Based on the obtained value and on the adopted response the node is replaced by using production π_2 with probability 0.2. Notice the semantics of such response: a node D (defector) is replaced by two nodes C (cooperator). The parallel step of $\mathcal{B}$ is completed when all the nodes in the network (a) have been considered. The new network (g) is then obtained.

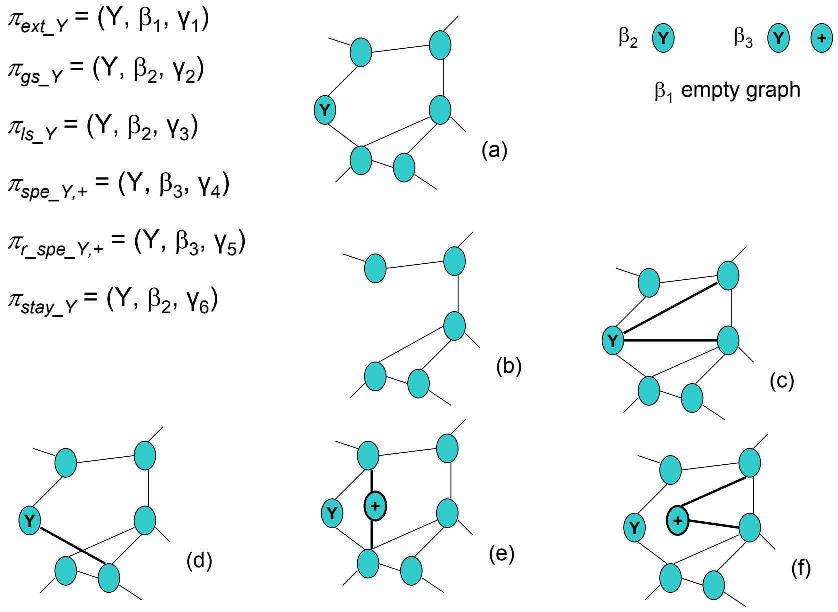

π_{ext_Y} = (Y, β1, γ1)
π_{gs_Y} = (Y, β2, γ2)
π_{ls_Y} = (Y, β2, γ3)
$\pi_{spe\_Y,+}$ = (Y, β3, γ4)
$\pi_{r\_spe\_Y,+}$ = (Y, β3, γ5)
π_{stay_Y} = (Y, β2, γ6)
β2 Y
β3 Y +
β1 empty graph
Y
(a)
(b)
Y
(c)
Y
(d)
Y +
(e)
Y +
(f)

Figure 7: A food-web can be represented as a graph, where nodes are species and edges denote the interactions between (partner) species. Each node is labelled by + and − that denotes if the species is positive or negative. A part of a more complex food-web is shown in (a); in particular, there is a species "labelled" by Y (can be arbitrarily instanced as + or −) interacting with other two species. We consider four types of dynamics for the food-web: a species can survive – it simply keeps the same interactions partners; a species can change some of its partners; a (mother) species can speciate by creating a (daughter) species; a species can get extinct. These dynamics can be implemented by opportune productions that are sketched in the figure. The left side of the productions is constituted by the label Y of the species to be replaced, that can be either + or −. The right side can consist of a graph with two species (in case of speciation), with one species (in case of survival and change of partners), or can be the empty graph (in case of extinction). The embedding functions for the different productions can be understood from the Figure. The *survival of a species*, not shown here, is implemented by the production π_{stay_Y}. In this case the embedding function γ_6 must keep all connections - simply, no change is done. In case of *speciation*, we have two possible dynamics: (i) the daughter species interacts with two species randomly selected, or (ii) the daughter species interacts with all the species that are partners of the mother. In both cases, the daughter species is positive. These two types of speciation can be implemented by productions (i) $\pi_{r\_spe\_Y,+}$ and (ii) $\pi_{spe\_Y,+}$, respectively. The embedding functions γ_5 and γ_4 of these productions can be understood from the Figure: $(a)-(f)$ represents the application of $\pi_{r\_spe\_Y,+}$ while $(a)-(e)$ represents the application of $\pi_{spe\_Y,+}$ (inserted nodes and added connections are in bold). A *species can change its partners* according to two possible dynamics: (i) the species changes one of its partners to one its neighbors, or (ii) the species has two more partners, searched randomly in the food-web. These two dynamics can be implemented by productions (i) π_{ls_Y} and (ii) π_{gs_Y}, respectively. The embedding functions of these productions can be understood from the Figure: $(a)-(c)$ describes the application of π_{gs_Y}, while $(a)-(d)$ describes the application of π_{ls_Y} (inserted connections are in bold). A *species can also get extinct*: implemented by production π_{ext_Y}; the embedding function

$\mathbf{\Gamma}_{\mathbf{1}} = \{\pi_{\mathrm{ls\_Y,}}, \pi_{\mathrm{ext\_Y,}}$
$\pi_{\mathrm{stay\_Y,}} \pi_{\mathrm{spe\_Y,+}}$
for Y = +,-}

$\eta_{\mathbf{1}}$:

x ≤0
$\eta^{x}_{Y}\ (\pi_{\mathrm{ext\_Y}}) = 1$
$\eta^{x}_{Y}\ (\pi) = 0$, all other π in Γ_1

0≤ x ≤2
$\eta^{x}_{Y}\ (\pi_{\mathrm{ls\_Y}}) = 1$
$\eta^{x}_{Y}\ (\pi) = 0$, all other π in Γ_1

2≤ x ≤4
$\eta^{x}_{Y}\ (\pi_{\mathrm{stay\_Y}}) = 1$
$\eta^{x}_{Y}\ (\pi) = 0$, all other π in Γ_1

4<x
$\eta^{x}_{Y}\ (\pi_{\mathrm{spe\_Y,+}}) = 1$
$\eta^{x}_{Y}\ (\pi) = 0$, all other π in Γ_1

$\mathbf{\Gamma}_{\mathbf{2}} = \{\pi_{\mathrm{gs\_Y,}}, \pi_{\mathrm{ext\_Y,}}$
$\pi_{\mathrm{stay\_Y,}} \pi_{\mathrm{r\_spe\_Y,+}}$
for Y = +, -}

$$\begin{array}{c|cc} & + & - \\ \hline + & c_{++} & c_{+-} \\ - & c_{-+} & c_{--} \end{array}$$

$\eta_{\mathbf{2}}$:

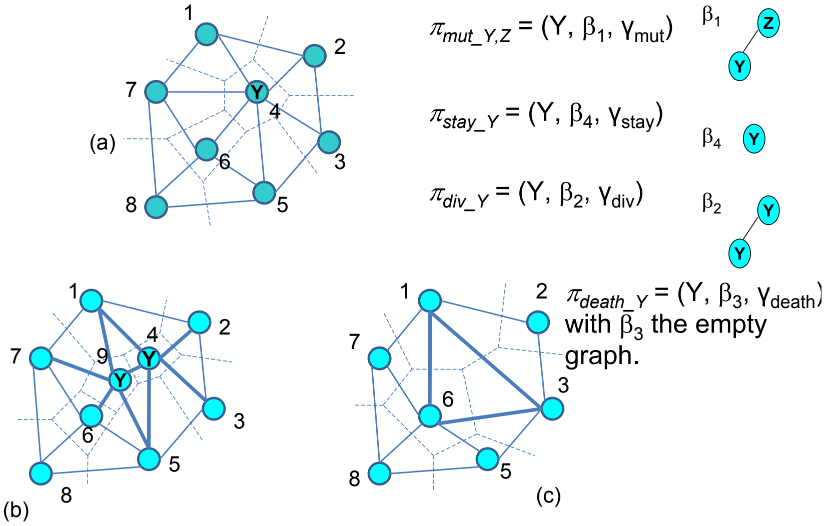
x ≤0
$\eta^{x}_{Y}\ (\pi_{\mathrm{ext\_Y}}) = 1$
$\eta^{x}_{Y}\ (\pi) = 0$, all other π in Γ_2

0≤ x ≤2
$\eta^{x}_{Y}\ (\pi_{\mathrm{gs\_Y}}) = 1$
$\eta^{x}_{Y}\ (\pi) = 0$, all other π in Γ_2

2≤ x ≤4
$\eta^{x}_{Y}\ (\pi_{\mathrm{stay\_Y}}) = 1$
$\eta^{x}_{Y}\ (\pi) = 0$, all other π in Γ_2

4<x
$\eta^{x}_{Y}\ (\pi_{\mathrm{r\_spe\_Y,+}}) = 1$
$\eta^{x}_{Y}\ (\pi) = 0$, all other π in Γ_2

Figure 8: We sketch some of the components of a DSG that models a food-web with generalists and specialists positive and negative species. The DSG is executed starting from an *initial network (graph)*, not shown here, representing a food-web, with an arbitrary number of species (nodes). At each step of the DSG the graph obtained represents the current food-web: nodes correspond to positive and negative species, distinguished by their labels, as discussed in Figure 7. The entries of the payoffs matrix are used to quantify the interactions between positive and negative species. Since the payoffs represents various pairwise relationships, we suppose to use a node-additive value function: the value obtained by a species depends on the adjacent species (partners) and on the payoffs matrix. Moreover, some of the nodes are specialists while the rest are generalists. We assume that specialists are the nodes that use nodal responses (Γ_1, η_1) while generalists are the nodes that use nodal response (Γ_2, η_2). The productions in Γ_1 and Γ_2 implement speciation, extinction, survival and change of partners as described in Figure 7. The functions η_1 and η_2 specify when and how to apply the productions. Essentially, η_1 and η_2 give the differences between specialists and generalists. For instance, if a specie gets a value x smaller than 0, then the specie gets extinct, with probability 1, in both cases. If it gets a value x between 0 and 2, a specialist uses, with probability 1, production π_{ls_Y}, so it re-patterns his links by doing a local search, while for the same range of values a generalist uses, with probability 1, the production π_{gs_Y}, so it re-patterns his links by doing a global search.



1
2
7
Y
4
(a)
6
3
8
5
$\pi_{mut\_Y,Z}$ = (Y, β1, γmut)
β1
Z
Y
π_{stay_Y} = (Y, β4, γstay)
β4
Y
π_{div_Y} = (Y, β2, γdiv)
β2
Y
Y
1
2
4
7
9
Y
Y
6
3
5
8
(b)
1
2
7
6
3
8
5
(c)
π_{death_Y} = (Y, β3, γdeath)
with β3 the empty graph.

Figure 9: A triangulated labeled graph representing a 2D cellular tissue, [5]: each node corresponds to a cell, cell-to-cell contact corresponds to a connecting edge and the dual graph represents the tissue. Cell division and cell death is presented (the operations are described in details in [5]). Cell mutation means that a daughter cell has a different label than the mother cell. Cell staying (rest), not shown in the Figure, means that the cell and its connections are kept unchanged and it represents the behavior of resting cells. Cell division, cell mutations, cell rest and cell death can be implemented by the sketched productions $\pi_{div}, \pi_{mut}, \pi_{stay}, \pi_{death}$, respectively. The left side of the productions is constituted by the label Y of the node to be rewritten. The right side can consist of a graph with two nodes (in case of division), with one node (in case of rest), or be the empty graph (in case of death). The embedding functions for the different productions can be understood from the Figure. In case of *cell rest*, the embedding function must keep all connections as the mother node. The function that implements such embedding is referred as γ_{stay}. In the case of *cell division*, $(a) - (b)$, the two created nodes must be connected in a random manner to the adjacent nodes of the mother, with the restriction to keep the graph triangulated. The edges that are added are in bold. The function that implements such embedding is referred as γ_{div} (the embedding γ_{mut} works in the same manner). *Cell death* is presented in $(a) - (c)$. In this case, a node is removed and the connections between the adjacent nodes of the deleted node must be modified accordingly, to keep the graph triangulated, as shown in $(a) - (c)$. The function that implements such embedding is referred as γ_{death}. Notice that, in this last case, the embedding function is more complex that the ones presented in Section 1.2 and must "invoke" the rewriting/replacement of nodes adjacent to the mother node. The productions using such type of embedding have been considered in literature under the name of ordered/programmed productions. The presentation in details of these types of productions is not in the the scope of this paper, but the reader can find their descriptions in [9], in the survey on graph transformation [17] and [13].

Nodal response: (Γ, η)

$$\begin{aligned}\Gamma = \{&\pi_{div\_Y}, \pi_{death\_Y}, \pi_{stay\_Y} \textit{ for } Y = N, C_1, C_2, T\} \\ \cup\{&\pi_{mut\_Y,Z} \textit{ for } Y = N \textit{ and } Z = C_1, C_2 \textit{ or } \\ &Y = C_1, C_2 \textit{ and } Z = T\}\end{aligned}$$

η : π_{death_Y} has an higher probability when $Y = C_1, C_2$ than when $Y = N, T$

game

$$\begin{array}{lllll} & N & C_1 & C_2 & T \\ N & c_{NN} & c_{N1} & c_{N2} & c_{NT} \\ C_1 & c_{1N} & c_{11} & c_{12} & c_{1T} \\ C_2 & c_{2N} & c_{21} & c_{22} & c_{2T} \\ T & c_{TN} & c_{T1} & c_{T2} & c_{TN} \end{array}$$

Initial network

Value functions: node-additive

Figure 10: We present some of the components of a DSG that can model cooperation in tumor development. The DSG is only sketched with the main goal to illustrate the intuitions behind the model. The DSG is executed starting from an *initial network* that is composed by a single node. At each step of the DSG the network obtained represents a tissue: nodes correspond to cells, as discussed in Figure 9 – we use "cells" instead of "nodes". Each cell has a *label* that corresponds to its *state* – we use "state" instead of "label". We consider *four possible states* for the cells: N (normal), C_1 and C_2 (partially mutated), and T (independent). The strategies of the proposed game are the possible four states of the cells. The value function used by the DSG is node-additive (the payoffs abstract quantities of growth factors). This means that the value of a cell depends on the adjacent cells and on the payoffs matrix. In this way, the entries $c_{11}, c_{12}, c_{21}, c_{22}$ of the payoffs matrix represent the amount of cooperation between partially mutated cells. In the proposed DSG there is only one nodal response and is the pair (Γ, η) (hence, all cells divide, mutate, rest and die as determined by such nodal response). Specifically, Γ is the set of productions that implement cells division, cell death, cell mutation and cell rest (as described in Figure 9) for the cells in the four possible states N, C_1, C_2, T (we suppose that only cells in state N, C_1, C_2 can mutate). The function η (very informally sketched here) should be instanced according to the known division and mutation rates for living cells. In particular, the function η must be defined in such a way that: (i) for an arbitrary state Y, a cell with state Y has a probability to rest (i.e., be replaced by using π_{stay_Y}) proportional to the obtained value; (ii) cells in states C_1 or C_2 should have higher chance to die (i.e., be replaced by π_{death_Y}) than cells in states N or T. In other words, the function η should model the fact that (i) higher values means for the cells higher probability to rest (and to not die) and that (ii) partially mutated cells are not independent and have higher chance to die than normal or independent cells. One can investigates the hypothesis of cooperation in tumor development, [3], by tuning the entries of the game and looking to the types of tissues (networks) produced/generated during the DSG.