Markov Chain Based Analysis of Agent-Based Immunological System

Aleksander Byrski[1], Robert Schaefer[1], and Maciej Smołka[2]

[1] Dept. of Computer Science, AGH University of Science and Technology
Al. Mickiewicza 30, 30-059 Kraków, Poland
{olekb,schaefer}agh.edu.pl
[2] Institute of Computer Science, Jagiellonian University,
ul. Łojasiewicza 6, 30-348 Kraków, Poland
smolka@ii.uj.edu.pl

Abstract. In the course of the paper we recall the Markov model for immunological Evolutionary Multi-Agent System. The model allows to study dynamic features of the computation and increases understanding the considered classes of systems. The main contribution of the paper is the draft of the proof of the ergodicity feature of the Markov chain modelling iEMAS dynamics.

1 Introduction

Certain heuristic system may never become one ultimate answer to solving all possible optimisation problem [22]. On the other hand, when building complex hybrid algorithms, an important question should be posed, does this system is able to work at all? This is important, because complex search methods may affect the ability to find all possible answers to the given problem, therefore formal proving of certain features of the computation becomes an important argument in the discussion of applicability of certain search methods.

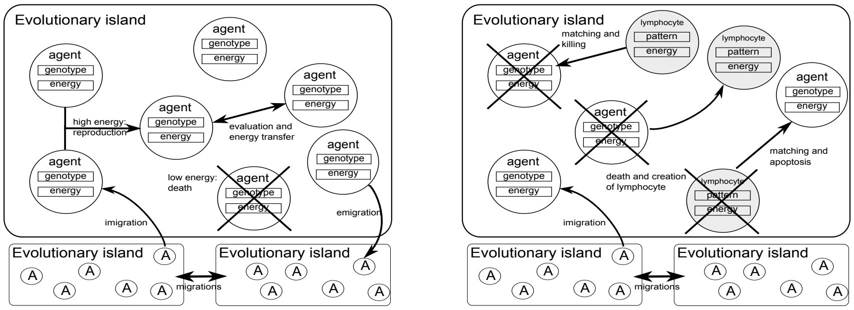
The formal model presented by Vose [20] proved in the most simple, yet effective way, asymptotic guarantee of success, i.e. "ability to find all local maximizers (minimizers) with probability 1 after infinite number of epochs" [18, 12, 15] in the analysis of the Simple Genetic Algorithm (SGA) behaviour, formally confirming the possibility of using SGA for global optimisation. Formal models for genetic algorithms were also proposed by other researchers, providing a deeper insight into the long term, steady state behaviour of large population EAs [9, 19, 16] or modelling specific features of EAs such as selection, genetic drift, niching etc. [10, 14, 11]. Many other, more complex, biologically-inspired computational techniques were proposed (e.g. memetic systems, immune-inspired systems), however, the problem of construction of appropriate mathematical models and approaches at proving asymptotic guarantee of success do not seem to be studied extensively or even were not undertaken at all.

In the course of paper we recall the basic features of the stochastic Markov models already introduced in the works of Byrski et al. (e.g. [5, 17, 6]). We define the space of states, synchronisation mechanism, and we draw the probability transition function. The main contribution of the paper is a draft of the sequence of actions proving the ergodicity of Markov chain constructed for iEMAS (immunological Evolutionary Multi Agent System introduced by Byrski and Kisiel-Dorohinicki [2]) by transferring the system between two arbitrarily chosen states. The formal proof for EMAS (predecessor of iEMAS, being a general optimisation system leveraging paradigms of evolutionary computation and agency, introduced by Cetnarowicz [7]) ergodicity has already been submitted for publication, we will follow with full proof of ergodicity of iEMAS in the near future.

2 Evolutionary and Immunological Agent-Based Computation

EMAS and iEMAS are general-purpose optimisation systems leveraging paradigms of evolutionary computation and agency, following work of Cetnarowicz [7]) that has already proven its efficiency for certain class of problems (see e.g., [4, 2, 3]).

In the simplest possible model of an evolutionary multi-agent system there is one type of agents and one resource defined. Genotypes of agents represent feasible solutions to the problem.



(a) EMAS

(b) iEMAS

Fig. 1. Evolutionary (EMAS) and immunological (iEMAS) multi-agent system

Energy is exchanged by agents in the process of evaluation. The agent increases its energy when it finds out that one (e.g. randomly chosen) of its neighbours, has lower fitness. In this case, the agent takes part of its neighbour's energy, otherwise, it passes part of its own energy to the evaluated neighbour. The level of life energy triggers actions of death and reproduction (low energy causes death while high energy makes reproduction possible). Attaining predefined level of energy may lead an agent also to migrate from one evolutionary island to another (see Fig. 1(a)).

Immune-inspired approaches were applied to many problems, such as classification or optimisation (e.g. [8]). The most often used algorithms of clonal and negative selection correspond to their origin and are used in a variety of applications [21].

The main idea of applying immunological inspirations to speed up the process of selection in EMAS is based on the assumption that 'bad' phenotypes come from 'bad' genotypes. Thus, a new group of agents (acting as lymphocyte T-cells) may be introduced [3]. They are responsible for recognising and removing agents with genotypes similar to the genotype pattern possessed by these lymphocytes. Another approach may introduce specific penalty applied by T-cells for recognised agents (certain amount of the agent's energy is removed) instead of removing them from the system. The general structure of iEMAS (immunological EMAS) is presented in Fig. 1(b).

Of course there must exist some predefined affinity (lymphocyte-agent matching) function, which may be based, e.g., on the percentage difference between corresponding genes. Agents-lymphocytes are created in the system after the action of death. The late agent genotype is transformed into lymphocyte patterns by means of mutation operator, and the newly created lymphocyte (or group of lymphocytes) is introduced into the system.

In both cases, new lymphocytes must undergo the process of negative selection. In a specific period of time, the affinity of immature lymphocytes' patterns to 'good' agents (possessing relatively high amount of energy) is tested. If it is high (lymphocytes recognize 'good' agents as 'non-self') they are removed from the system. If the affinity is low, it is assumed that they will be able to recognize 'non-self' individuals ('bad' agents) leaving agents with high energy intact. The life span of lymphocytes is controlled by specific, renewable resource (strength), used as a counter by the lymphocyte agent (see Fig. 1(b)).

3 Agent-based management and synchronisation

We start considerations from evolutionary multi-agent systems solving global optimisation problems (cf. [17]), which consist in finding all global minimizers of a given nonnegative fitness function over a finite genetic universum U with cardinality r. EMAS agents belong to a predefined finite set Ag. Every active agent is assigned to a location (evolutionary island) from the set $Loc = \{1, \ldots, s\}$. The locations are interconnected with channels along which agents can migrate. A channel topology is given by a symmetric relation $Top \subset Loc^2$.

Continuing considerations presented in e.g. [5, 17] we focus on the *Immunologically based Evolutionary Multi-Agent System* (iEMAS) that contains (besides dynamic collection of agents that belong to the predefined finite set Ag identical to the one of EMAS) a dynamic collection of lymphocytes that belong to the finite set Tc. Lymphocytes are unambiguously indexed by the genotypes from U, so that $\#Tc = \#U =$ r.

The lymphocytes have a similar structure as the agents previously defined, however, their actions differ (because their goals differ from the agents' goals) and their total energy does not have to be constant.

iEMAS may be modeled as the following tuple:

$$< U, \{P_i\}_{i \in Loc}, Loc, Top, Ag, \{agsel_i\}_{i \in Loc}, locsel, \{LA_i\}_{i \in Loc}, MA, \omega, Act, \\ \{typesel_i\}_{i \in Loc}, \{tcsel_i\}_{i \in Loc}, Tc, Tcact > \quad (1)$$

where:

- MA (master agent) is used to synchronize the work of the locations; it allows to perform actions in particular locations. This agent is also used to introduce necessary synchronisation into the system.
- $locsel : X \to \mathcal{M}(Loc)$ is the function used by MA to determine which location should be allowed to perform the next action,
- LA_i (local agent) is assigned to each location; it is used to synchronize the work of computational agents present in its location, LA_i chooses the computational agent and lets it evaluate a decision and perform the action, at the same time asking MA whether this action may be performed.
- $agsel_i : X \to \mathcal{M}(U \times P_i)$ is a family of functions used by local agents to select the agent that may perform the action, so every location $i \in Loc$ has its own function $agsel_i$.
- $\omega : X \times U \to \mathcal{M}(Act)$ is a function used by agents for selecting actions from the set Act; both these symbols will be described later.
- Act is a predefined, finite set of actions.
- $typesel_i$ is a function used to select the type of agent in i-th location to interact with the system in the current step,
- $tcsel_i$ is used to choose a lymphocyte in i-th location to interact with the system in the current step,
- φ is the decision function for lymphocytes,
- $Tcact$ is a set of actions that may be performed by lymphocytes.

Hereafter $\mathcal{M}(\Omega)$ shall stand for the space of probabilistic measures over Ω.

In order to design a Markov model of the system with relaxed synchronisation (i.e. such that agents present in different locations may act in parallel), a timing mechanism must be introduced, i.e. all state changes must be assigned to subsequent time moments $t_0, t_1, \ldots$.

In Fig. 2 the scheme of the synchronisation mechanism built using agents, $LA_i, i \in Loc$ and MA is presented.

The computational agent CA present in the location i in every observable time moment chooses an action it wants to perform and asks its supervisor (local agent LA_i) for a permission to carry on. Then it suspends its work waiting for the permission. When the permission is granted and the decision assigned to the considered action is positive, the computational agent changes the state of the location. Afterwards the agent suspends its work again in order to get a permission to perform a subsequent action. The immunological agent TC works in a similar way to CA, managing behaviour of a single lymphocyte.

The local agent LA_i receives signals containing actions to be performed from all its agents. Then chooses one computational agent which should try to perform its action. This action is reported to the master agent MA and after receiving

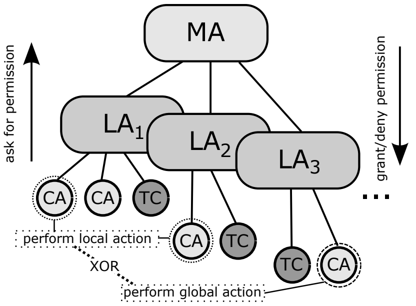


Fig. 2. Scheme of the synchronisation mechanism

permission, the computational agent can perform the action. All other agents are stopped from performing their actions.

The master agent MA waits for all requests from location and then chooses randomly one location. If this location asks for permission to perform global action, then it is granted this permission and all other locations are rejected. Otherwise all locations which asked for the permission to perform global action are rejected and all those asking for permission to perform local action — are granted.

4 System state

In this section we cite the description of EMAS state and extend it by adding a matrix describing iEMAS state (following [17]).

4.1 EMAS state

Let us introduce the set of three-dimensional, incidence and energy matrices $x \in X$ with s layers (corresponding to all locations) $x(i) = \{x(i, gen, n),\ gen \in U,\ n \in P_i\},\ i \in Loc$. The layer $x(i)$ will contain energies of agents in i-th location. In other words, if $x(i, gen, k) > 0$, it means that the k-th clone of the agent containing the gene $gen \in U$ is active, its energy equals $x(i, gen, k)$ and it is located in i-th location.

We introduce the following coherency conditions:

- $(\cdot, j, k)$-th column contains at most one value greater than zero, which expresses that the agent with k-th copy of j-th genotype may be present in only one location at a time, whereas other agents containing copies of j-th genotype may be present in other locations;
- incidence and energy matrices' entries are non-negative $x(i, j, k) \geq 0,\ \forall\ i = 1, \ldots, s,\ \ j = 1, \ldots, r,\ \ k = 1, \ldots, p$ and $\sum_{i=1}^{s} \sum_{j=1}^{r} \sum_{k=1}^{p} x(i, j, k) = 1$, which means that total energy contained in the whole system is constant, equal to 1;

- each layer $x(i)$ contains at most q_i values greater than zero, which denotes the maximum capacity of the i-th location, moreover, the quantum of energy Δe is lower or equal than total energy divided by the maximal number of individuals that may be present in the system $\Delta e \leq \frac{1}{\sum_{i=1}^{s} q_i}$ which allows us to achieve maximal population of agents in the system;
- reasonable values of p should be greater or equal to 1 and less or equal to $\sum_{i=1}^{s} q_i$; we assume that $p = \sum_{i=1}^{s} q_i$ which assures that each configuration of agents in locations is available, respecting the constrained total number of active agents $\sum_{i=1}^{s} q_i$; increasing p over this value does not enhance the descriptive power of the presented model;
- the maximal number of copies for each location $\#P_i$ should not be less than q_i, because we want to allow a system state in which a particular location is filled with clones of one agent; obviously increasing $\#P_i$ over q_i is only a formal constraint relaxation, so finally we assume that $\#P_i = q_i$.

Gathering all these conditions, the set of three-dimensional incidence and energy matrices may be described in the following way.

$$\begin{gathered}\Lambda = \Big\{ ince \in \{0, \Delta e, 2 \cdot \Delta e, 3 \cdot \Delta e, \ldots, m \cdot \Delta e\}^{s \cdot r \cdot p}, \Delta e \cdot m = 1, \\ \sum_{i=1}^{s} \sum_{j=1}^{r} \sum_{k=1}^{p} x(i,j,k) = 1, \forall\ i = 1, \ldots, s:\ \sum_{j=1}^{r} \sum_{k=1}^{p} [x(i,j,k) > 0] \leq q_i, \qquad (2) \\ \forall\ i = 1, \ldots, s,\ j = 1, \ldots, r,\ k \notin P_i:\ x(i,j,k) = 0, \\ \forall j = 1, \ldots, r,\ k = 1, \ldots, p:\ \sum_{i=1}^{s} [x(i,j,k) > 0] \leq 1 \Big\}\end{gathered}$$

where $[\cdot]$ denotes the value of the logical expression contained in the parentheses.

4.2 iEMAS state

In addition to the EMAS state describing the location and energy of agents (see (4.1)), we need to consider a set of matrices containing similar information for lymphocytes. Yet there is no need to assure the constant total energy for lymphocytes. We describe this additional set of lymphocyte incidence and energy matrices in the following way:

$$\begin{gathered}\Gamma = \Big\{ tcince \in [0, \Delta e, \ldots, n \cdot \Delta e]^{r \cdot s} : \forall\ i = 1, \ldots, s\ \ \sum_{j=1}^{r} [tcince(i,j) > 0] \leq tcq_j \\ \text{and } \forall\ j = 1, \ldots, r\ \ \sum_{i=1}^{s} [tcince(i,j) > 0] \leq 1 \Big\} \qquad (3)\end{gathered}$$

where $tcince(i,j)$ stands for energy of tc_j being active in the location i. The integers tcq_j, $j = 1, \ldots, s$ stand for the maximum number of lymphocytes in particular locations. It is most convenient to assume $tcq_j = q_j$, $\forall j = 1, \ldots, s$.

The space of iEMAS states is defined as follows:

$$X = \Lambda \times \Gamma \qquad (4)$$

5 System behaviour

Let us denote by X_{gen} the subset of states in which there are active agents with genotype $gen \in U$ or an active lymphocyte.

5.1 EMAS behaviour

Each action $\alpha \in Act$ will be represented as the pair of function families $(\{\delta_\alpha^{gen}\}_{gen \in U}, \{\vartheta_\alpha^{gen}\}_{gen \in U})$. The functions

$$\delta_\alpha^{gen} : X \to \mathcal{M}(\{0,1\}) \tag{5}$$

represent the decision to be taken: whether the action can be performed or not. The action α is performed with the probability $\delta_\alpha^{gen}(x)(1)$ by the agent $ag_{gen,n}$ at the state $x \in X$ and rejected with the probability $\delta_\alpha^{gen}(x)(0)$.

Next, the formula

$$\vartheta_\alpha^{gen} : X \to \mathcal{M}(X) \tag{6}$$

defines the non-deterministic state transition functions, so that ϑ_α^{gen} is caused by the execution of the action α by the agent $ag_{gen,n}$. Because the function is invoked only if the agent in active, it is enough to define its restriction $\vartheta_\alpha^{gen}|X_{gen}$ and take an arbitrary value on $X \setminus X_{gen}$.

If any action is rejected, the trivial state transition

$$\vartheta_{null} : X \to \mathcal{M}(X) \tag{7}$$

such that for all $x \in X$

$$\vartheta_{null}(x)(x') = \begin{cases} 1 & \text{if } x = x' \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

is performed.

The probability transition function for the action α performed by the agent containing the genotype gen

$$\varrho_\alpha^{gen} : X \to \mathcal{M}(X) \tag{9}$$

is given by the formula

$$\begin{aligned} \varrho_\alpha^{gen}(x)(x') = \delta_\alpha^{gen}(x)(0) \cdot \vartheta_{null}(x)(x') \\ +\delta_\alpha^{gen}(x)(1) \cdot \vartheta_\alpha^{gen}(x)(x') \end{aligned} \tag{10}$$

where $x \in X$ denotes a current state and $x' \in X$ a consecutive state resulted from the conditional execution of α.

5.2 iEMAS behaviour

We introduce the function $typesel_i$ choosing which type of agents will have the possibility of performing the action:

$$typesel_i : X \rightarrow \mathcal{M}(\{0,1\}) \tag{11}$$

when 0 is chosen, one of the agents is activated, when 1 — the lymphocyte.

The function choosing which agent will be activated $agsel_i$ is like in EMAS but it now depends in some way on the extended state from X defined by (4). Now we introduce a new function that will choose which lymphocyte will be activated:

$$tcsel_i : X \rightarrow \mathcal{M}(Tc) \tag{12}$$

The function ω choosing the action for the active agent remains intact, though its domain changes (because of the new state definition, see (4)).

The function choosing the action for the active lymphocyte is the following:

$$\varphi : U \times X \rightarrow \mathcal{M}(Tcact) \tag{13}$$

We will use the family of functions $\eta_\alpha^{gen} : X \rightarrow \mathcal{M}(X)$ where $gen \in U$, $\alpha \in Tcact$. Each of them expresses the probability transition imposed by the lymphocyte tc_{gen} that performs the action $\alpha \in Tcact$. They are given by the general formula:

$$\eta_\alpha^{gen}(x)(x') = \gamma_\alpha(gen,x)(\{0\}) \cdot \vartheta_{null}(x)(x') + \gamma_\alpha(gen,x)(\{1\}) \cdot \kappa_\alpha^{gen,n}(x)(x') \tag{14}$$

The agents' and lymphocytes' actions may be divided into two distinct types: global — they change the state of the system in two or more locations, so only one global action may be performed at a time, local — they change the state of the system inside one location respecting only the state of local agents, only one local action for one location may be performed at a time.

Therefore we divide the Act set in the following way: $Act = Act_{gl} \cup Act_{loc}$ and accordingly, $Tcact : Tcact = Tcact_{gl} \cup Tcact_{loc}$. Speaking informally, local actions (elements of $Act_{loc}, Tcact_{loc}$) change only the entries of the layer $x(i)$ of the incidence and energy matrices if the location $i \in Loc$ contains the agent performing a certain action. Moreover, these actions do not depend on other layers of x. The action *null* is obviously "the most local one", because it does not change anything at all.

In the case of EMAS and iEMAS, actions such as evaluation or lymphocyte pattern matching may be perceived as local, at the same time action of migration is perceived as global. The above-stated conditions may be defined formally and may be used to prove commutativity of iEMAS (cf. [5, 17]), however here we skip this proof because lack of space.

6 Parallel iEMAS dynamics

At the observable moment at which EMAS takes the state $x \in X$ all agents in all locations notify their local agents their intent to perform an action, all local agents choose an agent with the distribution given by the $agsel_i(x)$, $i \in Loc$ function and then notify the master agent of their intent to let perform an action by one of their agents. The master agent chooses the location with the probability distribution given by $locsel(x)$.

We extend the model of EMAS dynamics in order to model the behaviour of iEMAS. The probability that in the chosen location $i \in Loc$ the agent or lymphocyte wants to perform local action is as follows:

$$\xi_i(x) = typesel(x)(\{0\}) \sum_{gen \in U} \sum_{n=1}^{p} (agsel_i(x)(\{gen, n\}) \\ \cdot \omega(gen, x)(Act_{loc})) + typesel(x)(\{1\}) \quad (15)$$

The probability that the master agent will chose the location with the agent intending to perform the local action is:

$$\zeta^{loc}(x) = \sum_{i \in Loc} locsel(x)(\{i\}) \xi_i(x) \quad (16)$$

of course the probability of choosing the global action by the master agent is:

$$(1 - \zeta^{loc}(x)) = \zeta^{gl}(x) \quad (17)$$

If the global action is chosen, the state transition is given by:

$$\tau^{gl}(x)(x') = \sum_{i \in Loc} locsel(x)(\{i\}) \cdot \left(\sum_{gen \in U} \sum_{n=1}^{p} agsel(x)(\{gen, n\}) \cdot \\ \left(\sum_{\alpha \in Act_{gl}} \omega(gen, x)(\{\alpha\}) \cdot \varrho_{\alpha}^{gen,n}(x)(x') \right) \right) \quad (18)$$

Let us state the set of action sequences containing at least one local action:

$$Act_{+1loc} = \left\{ (\alpha_1, \ldots, \alpha_s) \in (Act \cup Tcact)^s; \\ \sum_{i=1}^{s} [\alpha_i \in (Act_{loc} \cup Tcact)] > 0 \right\} \quad (19)$$

The probability that in i-th location the agent ag_{gen_i,n_i} or the lymphocyte $tc_{\widetilde{gen_i}}$ chooses the action α_i is given by:

$$\mu_{\alpha_i, gen_i, n_i, \widetilde{gen_i}}(x) = typesel(x)(\{0\}) \cdot agsel_i(x)(\{gen_i, n_i\}) \omega(gen_i, x)(\{\alpha_i\}) + \\ typesel(x)(\{1\}) tcsel_i(x)(\{\widetilde{gen_i}\}) \varphi(\widetilde{gen_i}, x)(\{\alpha_i\}) \quad (20)$$

Let us define a multi-index:

$$\begin{aligned} ind = (\alpha_1, \ldots, \alpha_s; (gen_1, n_1), \ldots, (gen_s, n_s);\ (\widetilde{gen_1}), \ldots, (\widetilde{gen_s})) \\ \in IND = (Act \cup Tcact)^s \times (U \times \{1, \ldots, p\})^s \times U^s \end{aligned} \tag{21}$$

the probability that in consecutive locations agents ag_{gen_i, n_i} or lymphocytes $tc_{\widetilde{gen_i}}$ will choose the actions α_i is given by:

$$\mu_{ind}(x) = \prod_{i=1}^{s} \mu_{\alpha_i, gen_i, n_i, \widetilde{gen_i}}(x). \tag{22}$$

Transition function for parallel system is following:

$$\tau^{loc}(x)(x') = \sum_{(\alpha_1,\ldots,\alpha_s) \in Act_{+1loc}} \sum_{ind \in IND} \mu_{ind}(x)(\pi_1^{ind}(x) \circ, \ldots, \circ\, \pi_s^{ind}(x))(x') \tag{23}$$

where π_i is defined as:

$$\pi_i^{ind}(x) = \begin{cases} \varrho_{\alpha_i}^{gen_i, n_i}(x), & \alpha_i \in Act_{loc} \\ \eta_{\alpha_i}^{\widetilde{gen_i}}(x), & \alpha_i \in Tcact \\ \vartheta_{null}, & \alpha_i \in Act_{gl} \end{cases} \tag{24}$$

The value of $(\pi_1^{ind}(x) \circ, \ldots, \circ\, \pi_s^{ind}(x))(x')$ does not depend on the composition order, because transition functions associated with local actions commutate pairwise (see 5.2) . Finally, we may derive the following observation.

Observation 1 *The probability transition function for the parallel iEMAS model is given by the formula*

$$\tau(x)(x') = \zeta^{gl}(x)\tau^{gl}(x)(x') + \zeta^{loc}(x)\tau^{loc}(x)(x') \tag{25}$$

and formulas (15) – (24).

Observation 2 *The stochastic state transition of iEMAS given by formula (25) satisfies the Markov condition.*

Proof. All transition functions and probability distributions given by formulas (15)–(24) depend only on the current state of the system, which motivates the Markovian features of the transition function τ given by (25). The transition functions do not depend on the number of step at which is applied what motivates the stationarity of the chain.

7 iEMAS ergodicity proof draft

In this section we present a draft of a proof of the ergodicity feature for the Markov chain describing the behaviour of iEMAS.

Theorem 1. *Assume that the following assumptions hold.*

1. *The migration energy threshold is lower than the total energy divided by the number of locations* $e_{migr} < \frac{1}{s}$*. This assumption ensures that there will be at least one location in the system in which an agent is capable of performing migration (by gathering enough energy from its neighbors).*
2. *The quantum of energy is lower than or equal to the total energy divided by the maximum number of agents that may be present in the system* $\Delta e \leqslant \frac{1}{\sum_{i=1}^{s} q_i}$ *. This assumption allows to achieve a maximal population of agents in the system.*
3. *Reproduction (cloning) energy is lower than two energy quanta* $e_{repr} \leqslant 2\Delta e$*.*
4. *The amount of energy passed from parent to the child during cloning action is equal to* Δe *(so* $n_1 = 1$*).*
5. *The maximum number of agents on every location is greater than one,* $q_i > 1, i = 1, \ldots, s$*.*
6. *Locations are totally connected, i.e.* $Top = Loc^2$*.*
7. *Each active agent can be selected by its local agent with strictly positive probability.*
8. *The families of probability distributions being the parameters of EMAS have uniform, strictly positive lower bounds.*

Then the Markov chain modeling iEMAS (see equation (25)) is irreducible, i.e. all its states communicate.

In order to prove the Theorem 1, it is enough to show that the passage from x_b to x_e (two arbitrarily chosen states from X) may be performed in a finite number of steps with probability strictly greater then zero.

Let us consider the following sequence of stages.

- **Stage 0**: In every location in parallel: If the location is full, an agent is chosen, and it performs sequentially evaluation action with one of its neighbors in order to remove it (to make possible incoming migration from any other location, in case this location is full). After removing one of its neighbors the agent tries to perform any global action, e.g., migration (and fails), until the end of the stage. Otherwise, the trivial null state transition is performed. Final state of the Stage 0 is denoted by x_{0e}.
- **Stage 1 a**: One location is chosen, at which the sum of agents' energy exceeds the migration threshold in the state x_{0e} (based on assumption 1 of Theorem 1 there must be at least one). Then one agent from this location ag_{gen_1,n_1} (possibly possessing the largest energy in the state x_{0e}) is chosen. This agent performs a sequence of evaluation actions in order to gather all energy from all its neighbors, finally removing them from the system (by bringing their energy to zero).
- **Stage 1 b**: If there are any lymphocytes on the current location, they perform killing action, one by one, on the agent ag_{gen_1,n_1}, failing to remove it from the system, until all lymphocytes are removed. In the end, only one agent is present in the location.

- **Stage 1 c**: Now this agent begins the first migration round in order to visit all locations and to remove the agents (overtaking their energy by performing multiple *get* actions) and remove all lymphocytes. This round is finished at location i_1. Now, agent ag_{gen_1,n_1} possesses the total energy of the system which equals 1. Final state of the Stage 1 is denoted by x_{1e}. Note, that the state matrix has only one positive entry $x_{1e}(i_1, gen_1, n_1) = 1$.
- **Stage 2 a**: The agent performs cloning action producing one of the agents (ag_{gen_2,n_2}) that will be present on the location i_2, one of the locations in the state x_e containing total energy greater than the migration threshold. Now it passes all of its energy to this newly produced agent, finally being removed from the system. The purpose of the Stage 2 is to ensure that the agent recreating the population at the last location i_2 will be one of the agents present on this location in the state x_e. Otherwise if i_2 is full in the state x_e, ag_{gen_1,n_1} could not recreate this population. If ag_{gen_1,n_1} is active at the location i_2 at the state x_e (i.e. $x_e(i_2, gen_1, n_1) > 0$), the Stage 2 may be omitted (in this case ag_{gen_1,n_1} takes the role of ag_{gen_2,n_2} in the following stages).
- **Stage 3**: Next, the agent ag_{gen_2,n_2} begins the second migration round (starting migration from the location i_1) visiting all locations. In every visited location it performs cloning action producing one of the agents that will be present on this location in the state x_e. The cloned agent on each non-empty location (denoted by $ag_{gen_i^{first},n_i^{first}}$) will receive the total energy that should be assigned to its location, by the sequence of evaluation actions. The agent finishes the migration after recreating the population on the location i_2 (one of the islands containing a total energy in the state x_e greater than the migration threshold).
- **Stage 4 a**: In the system, the following sequence of actions assigned with the consecutive locations labeled by $i \in Loc$, non empty in the state x_e, is performed: every agent $ag_{gen_i^{first},n_i^{first}}$ perform a cloning action to produce an agent with the genotype of one of lymphocytes existing in the location in the state x_e. Now it performs a sequence of evaluation action to remove the agent (and the appropriate lymphocyte is performed). The lymphocyte performs a sequence of energy lowering actions to adjust its energy to the level observed in the state x_e. This is repeated until all the lymphocytes present in x_e are recreated.
- **Stage 4 b**: In the system, the following sequence of actions assigned with the consecutive locations labeled by $i \in Loc$, non empty in the state x_e, is performed: every agent $ag_{gen_i^{first},n_i^{first}}$ performs a sequence of cloning actions, recreating the population of agents on its location in the state x_e.
- **Stage 5**: In every location in parallel: agent $ag_{gen_i^{first},n_i^{first}}$ performs a sequence of evaluation actions with its neighbors in order to pass to them a sufficient amount of energy, required in the state x_e.

In the extended version of this paper we will show that every of aforementioned stages requires performing at most finite number of Markov chain steps by estimating their upper bounds. Moreover, we will show, that every aforementioned sequences have non-zero probabilities by estimating its lower bounds.

Theorem 1 leads us straightforwardly to the statement that every possible state of iEMAS is reachable (with positive probability) after performing a finite sequence of transitions independently on the initial population. We can reformulate such a conclusion in the following corollary.

Corollary 1. *All states containing the extrema are reachable from an arbitrary initial state. Thus iEMAS satisfies asymptotic guarantee of success in the sense of [18, 12, 15].*

The following theorem shows an additional feature of the considered Markov chain.

Theorem 2. *If the assumptions of Theorem 1 hold, then the Markov chain modeling EMAS is aperiodic.*

Proof. Let us consider a state of the chain such that every location contains a single computational agent only. In this case let us assume that each agent chooses evaluation as its next action. Because all agents have chosen local actions, the master agent will allow them all to perform their actions, however the absence of neighbors will force all the agents to perform the trivial (i.e. null) action. The transition probability function is then the s-fold composition of ϑ_{null}. Therefore in this case the system will return to the same state in one step. The probability of such transition is greater than zero. It means that the considered state is aperiodic. Our chain is irreducible (see Theorem 1) and therefore it has only one class of states, the whole state space, which obviously contains the considered aperiodic state. On the other hand, from Theorem 2.2 of [13] we know that aperiodicity is a state class property. In our case it means that all states of EMAS are aperiodic, which concludes the proof.

The following corollary is a consequence of Theorems 1 and 2.

Corollary 2. *The Markov chain modeling EMAS is* ergodic*.*

Remark 1. It is worth noticing that the Markov chain (25) is ergodic in a strong sense (not only irreducible, but also aperiodic). Such chains are quite often called *regular* (see e.g. [13]).

Because the space of states X is finite we may introduce the probability transition matrix:

$$Q = \{\tau(x)(y)\},\ x, y \in X \tag{26}$$

where τ is the iEMAS probability transition function — see Eq. (25). The Markov chain describing the iEMAS dynamics is a sequence of random variables (or, equivalently, probability distributions) $\{\xi_t\} \subset \mathcal{M}(X), t = 0, 1, \ldots$ where ξ_0 should be a given initial probability distribution. Of course we have that

$$\xi_{t+1} = Q \cdot \xi_t,\ t = 0, 1, \ldots \tag{27}$$

Remark 2. From Theorems 1 and 2 as well as the ergodic theorem [1] there exists a strictly positive limit $\widehat{\xi} \in \mathcal{M}(X)$ (i.e., $\widehat{\xi}(x) > 0, \forall\ x \in X$) of the sequence $\{\xi_t\}$ as $t \to +\infty$. This equilibrium distribution does not depend on the initial probability distribution ξ_0.

8 Conclusions

In the course of this contribution, a formal model for iEMAS has been recalled and adjusted for a discrete system state space. The space of states and the transition functions allowing to construct a uniform Markov chain model have been proposed. This model based on stationary Markov chains allows a better understanding of the behaviour of the proposed complex systems as well as their constraints.

One of the main implications of the analysis conducted here is the formulation and proof draft of Theorem 1 stating that the Markov chain based model of EMAS is stationary and ergodic. This will lead to an important conclusion stated in Corollary 1, namely that EMAS possesses the feature of asymptotic guarantee of success.

Ergodicity of Markov chain modelling iEMAS proves that this hybridization does not hamper the capabilities of solving optimization problem in general, the experimental results prove, that for certain problems with complex fitness function (e.g., evolution of neural network parameters) employing iEMAS is especially advantegeous.

A full formal proof for EMAS ergodicity has already been formulated and submitted for publication. In the near future we will follow with preparing a similar proof of iEMAS ergodicity.

References

1. Billingsley, P.: Probability and Measure. Wiley-Interscience (1995)
2. Byrski, A., Kisiel-Dorohinicki, M.: Immunological selection mechanism in agent-based evolutionary computation. In: Proc. of IIS: IIPWM '05 conference : Gdansk, Poland. Advances in Soft Computing, Springer (2005)
3. Byrski, A., Kisiel-Dorohinicki, M.: Agent-based evolutionary and immunological optimization. In: Computational Science - ICCS 2007, 7th International Conference, Beijing, China, May 27 - 30, 2007, Proceedings. Springer (2007)
4. Byrski, A., Kisiel-Dorohinicki, M., Nawarecki, E.: Agent-Based Evolution of Neural Network Architecture. In: Hamza, M. (ed.) Proc. of the IASTED Int. Symp.: Applied Informatics. IASTED/ACTA Press (2002)
5. Byrski, A., Schaefer, R.: Stochastic model of evolutionary and immunological multi-agent systems: Mutually exclusive actions. Fundamenta Informaticae 95(2-3), 263–285 (2009)
6. Byrski, A., Schaefer, R., Smołka, M.: Asymptotic features of parallel agent-based immunological system. In: Burczyński, T., Kołodziej, J., Byrski, A., Carvalho, M. (eds.) Proc. of 25th European Conference on Modelling and Simulation (2011)

7. Cetnarowicz, K., Kisiel-Dorohinicki, M., Nawarecki, E.: The application of evolution process in multi-agent world (MAW) to the prediction system. In: Tokoro, M. (ed.) Proc. of the 2nd Int. Conf. on Multi-Agent Systems (ICMAS'96). AAAI Press (1996)
8. Dasgupta, D., Nino, L.: Immunological Computation Theory and Applications. Auerbach (2008)
9. Davis, T.E., Principe, J.C.: A simulated annealing like convergence theory for the simple genetic algorithm. In: Proc. of the Fourth International Conference on Genetic Algorithms. pp. 174–181. San Diego, CA (1991)
10. Goldberg, D., Segrest, P.: Finite Markov chain analysis of genetic algorithms. In: Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application. pp. 1–8. L. Erlbaum Associates Inc., Hillsdale, NJ, USA (1987)
11. Horn, J.: Finite Markov Chain Analysis of Genetic Algorithms with Niching. In: Proceedings of the Fifth International Conference on Genetic Algorithms. pp. 110–117. Morgan Kaufmann (1993)
12. Horst, R., Pardalos, P.: Handbook of Global Optimization. Kluwer (1995)
13. Iosifescu, M.: Finite Markov Processes and Their Applications. John Wiley and Sons (1980)
14. Mahfoud, S.: Finite Markov Chain Models of an Alternative Selection Strategy for the Genetic Algorithm. Complex Systems 7, 155–170 (1991)
15. Rinnoy Kan, A., Timmer, G.: Stochastic global optimization methods. Mathematical Programming 39, 27–56 (1987)
16. Rudolph, G.: Massively parallel simulated annealing and its relation to evolutionary algorithms. Evolutionary Computation 1, 361–383 (1994)
17. Schaefer, R., Byrski, A., Smołka, M.: Stochastic model of evolutionary and immunological multi-agent systems: Parallel execution of local actions. Fundamenta Informaticae 95(2-3), 325–348 (2009)
18. Schaefer, R.: Foundations of global genetic optimization. Springer Verlag (2007)
19. Suzuki, J.: A Markov Chain Analysis on a Genetic Algorithm. In: Forrest, S. (ed.) Proc. of the 5th ICGA. pp. 146–154. Morgan Kaufmann (1993)
20. Vose, M.: The Simple Genetic Algorithm: Foundations and Theory. MIT Press, Cambridge, MA, USA (1998)
21. Wierzchoń, S.: Function optimization by the immune metaphor. Task Quaterly 6(3), 1–16 (2002)
22. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1(1), 67–82 (1997)