Fundamenta Informaticae XXI (2001) 1001–1024 1001

IOS Press

Stochastic Model of Evolutionary and Immunological Multi-Agent Systems: Parallel Execution of Local Actions

Robert Schaefer[1]

Aleksander Byrski[1]

Maciej Smołka[2]

[1]*Department of Computer Science*
AGH University of Science and Technology
Al. Mickiewicza 30, 30-059 Kraków, Poland
{schaefer,olekb}@agh.edu.pl

[2]*Institute of Computer Science*
Jagiellonian University
ul. Łojasiewicza 6
30-348 Kraków, Poland
smolka@ii.uj.edu.pl

Abstract. The refined model for the biologically inspired agent-based computation systems EMAS and iEMAS conforming to the BDI standard is presented. Moreover, their evolution is expressed in the form of the stationary Markov chains. This paper generalizes the results obtained by Byrski and Schaefer [7] to a strongly desired case in which some agent's actions can be executed in parallel. In order to find the Markov transition rule, the precise synchronization scheme allowing for the stepwise stochastic evolution of the system has to be established. The crucial feature which allows to compute the probability transition function in case of parallel execution of local actions is the commutativity of their transition operators. Some abstract conditions expressing such a commutativity which allow for easy classification of agent's actions as local or global ones are formulated and verified. The above-mentioned Markov model constitutes the basis of the asymptotic analysis of EMAS and iEMAS necessary to evaluate their search possibilities and efficiency.

Keywords: multi-agent systems, agent-based computation, stochastic modelling

1. Introduction

Evolutionary Multi-Agent Systems (EMAS) proposed in 1996 by Cetnarowicz [9] and later researched in [4, 2, 3, 12] try to enrich the classical evolutionary mechanisms using social inspirations [1]. The idea of EMAS was extended to the system iEMAS that involves also the immunologically inspired mechanisms (see [4, 2, 3]). The paper introduces the formal models of EMAS and iEMAS dynamics which express their evolution in the form of he stationary Markov chains.

Agents in EMAS may be perceived as autonomous individuals. Every agent is capable of observing its environment by gathering information which it finds important, making decisions which affect its activity and performing actions which lead to changes in the overall state of the system (see e.g. [10, 11]).

We will focus on system that solves the global optimization problems which consist of finding all global minimizers $\arg\min\{\Phi(x)\}, x \in \mathcal{D}$ of the objective: $\Phi : \mathcal{D} \to \mathbb{R}_+$ where $\mathcal{D} \subset \mathbb{R}^N$ stands for the admissible set of solutions. Every EMAS agent contains an immutable genotype, which stands for the encoded solution of the problem. Genotypes belong to the binary or real-number based genotype universum U.

Agents are assigned to locations (analogous of "islands", see e.g. [8]) and may migrate among them. Genetic operations performed on the agent's genotypes, such as crossover and mutation, are similar to those used in classical evolutionary algorithms.

Each agent is transformed asynchronously in the EMAS system. Selection mechanisms correspond to their prototype and are based on the existence of a non-renewable resource called *life energy*, which is gained and lost when agents perform actions [12]. Direct employment of different selection techniques (such as proportional or tournament-based) is impossible because of the asynchronous nature of the system and decomposition of the population. In order to enhance the possibilities of managing selection, especially to flexible change the selection pressure, the dedicated, agent-based techniques can be introduced. One such technique of enhancing EMAS consists of the introduction of a new group of agents acting as *lymphocytes* [2]. They are responsible for recognizing (assuming some predefined affinity function) and penalizing (by decreasing the agent's energy or even making it inactive) agents with genotypes similar to the pattern possessed. Thus immunological EMAS (iEMAS) is created [3].

This paper generalizes the results obtained by Byrski and Schaefer [7] to the strongly desired case in which some agent's actions can be executed in parallel. The agent's actions may fall into two separate classes: global and local. Global actions depend on and may change the whole state of the system, so their execution has to be mutually excluded. Each local action depends and may change only the state of the agents in the location of the agent that it currently executes, so they may be performed concurrently in separate locations.

In order to find the Markov transition rule, the precise synchronization scheme allowing for the stepwise stochastic evolution of the system have to be established. The additional governing agents and the mechanism of the blocking communication among them are introduced. These agents allow for executing single global action or at most one local action in each location in the single evolution step. The space of states and the definitions of actions as well as their transition operators described in [7] will be intensively utilized.

The main tool for obtaining the final formula is the Bayes theorem (as in the case of previous paper [7]) which allows to derive the formula of probability transition as the barycentric combination of elementary stochastic transformations implemented in particular actions. The crucial feature that allows to establish the probability transition function in case of parallel execution of local actions is commuta-

tivity of their transition operators. The abstract conditions of such commutativity which allow for easy classifying of agent's or T-cell's actions as local or global ones will be formulated and verified.

The Markov model mentioned above constitute the basis of the asymptotic analysis of EMAS ans iEMAS systems in order to evaluate their search possibilities and efficiency. It seems to be the way of verifying their probabilistic guarantee of success (e.g. by checking the ergodicity) and comparing with the other stochastic algorithms by comparing their limit invariant measures (if their exist).

2. EMAS definition

2.1. EMAS structure

EMAS contains a dynamic collection of agents that belong to the predefined finite set Ag one-to-one mapped on set $U \times \{1, \ldots, p\}$, where p is assumed maximum number of agents containing the same genotype. In other words, every agent $ag_{gen,n} \in Ag$ contains one potential solution of the given problem encoded as $gen \in U$, however, there may be more than one agent present in the system, containing this solution and the index $n \in \{1, \ldots, p\}$ is used to distinguish them. We restrict our considerations to the case of finite universa $\#U = r < +\infty$.

Active EMAS agents are contained in locations described by a set of immutable integer labels $Loc = \{1, \ldots, s\}$. The locations are linked by the channels along which agents may migrate from one location to another. The topology of channels is determined by the symmetric relation $Top \subset Loc^2$. We assume that the connection graph $\langle Loc, Top \rangle$ is coherent, and does not change during the system evolution.

The state of a single agent is characterized by the vector $\widetilde{ag}_{gen,n}$ from $[0, 1]^s$ such as at most one entry is strictly positive, which stands for the fraction of the total energy gathered by the agent. The position of the positive entry in $\widetilde{ag}_{gen,n}$ denotes the location from $\{1, \ldots, s\}$ in which the agent is present. When $\widetilde{ag}_{gen,n}$ is a zero vector it means that the agent is inactive and is not present in any location.

EMAS may be modeled as the following tuple:

$$< U, Loc, Top, Ag, \{agsel_i\}_{i \in Loc}, locsel, \{LA_i\}_{i \in Loc}, MA, \omega, Act > . \tag{1}$$

where:

MA (master agent) – it is used to synchronize the work of the locations; it allows to perform either one of global actions or all local ones signalled at a time. This agent is also used to introduce necessary synchronization 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_{loc} (local agent) – assigned to each location; it is used to synchronize the work of computational agents present in its location, LA_{loc} chooses the agent and let it evaluate the decision and perform the action at the same time asking MA whether this action may be performed.

$agsel_{loc} : X \to \mathcal{M}(Ag)$ is the family of functions used by local agents to select agent that may perform the action, so every location $loc \in Loc$ has its own function $agsel_{loc}$. The probability $agsel_{loc}(x)(\{gen\})$ vanishes when the agent $ag_{gen,n}$ is inactive in the state $x \in X$ or it is present in other location than loc,

ω is the function used by agents for selecting actions from the set Act, both these symbols will be described later.

Here and later $\mathcal{M}(\cdot)$ stands for the space of probabilistic measures.

2.2. EMAS state

Let us introduce the set of three-dimensional, incedence and energy matrices Λ (the same as in [7]) with s layers (corresponding to all locations) $ince(i) \in U \times \{1,\ldots,p\}, i = 1,\ldots,s$. The layer $ince(i)$ will contain energies of agents in i-th location a.e. $ince(i,gen,n) = \widetilde{ag}_{gen,n}(i)$. In other words, if $ince(i,gen,k) > 0$, it means that the k-th clone of the agent containing the gene $gen \in U$ is active, its energy equals $ince(i,gen,k)$ and it is located in i-th location.

Describing the space of states in e.g. [6] we assumed that there may be only one agent representing one unique genotype in all locations present in the system however now we relax this assumption making possible to create more than one agent in the system containing the same genotype (what is quite common in evolutionary computation, see e.g. [13]).

We need to introduce following assumptions and coherency conditions for the space of system states:

- each layer $ince(i)$ contains at most q_i values greater than zero, what denotes maximum capacity of the i-th location,
- reasonable values of p should be greater or equal to 1 and less or equal to $\sum_{i=1}^{s} q_i$. We assume the most convenient solution $p = \sum_{i=1}^{s} q_i$, by which each configuration of agents in locations is available, respecting the constrained total number of active agents $\sum_{i=1}^{s} q_i$. Increasing of p over this value does not enhance the descriptive power of the presented model,
- $(\cdot,j,k)$-th column (which makes $\widetilde{ag}_{j,k}$) contains at most one value greater than zero, what expresses that the agent with k-th copy of j-th genotype may be present in only one location at a time, of course other agents containing copies of j-th genotype may be present in other locations,
- incedence and energy matrix entries are non-negative $ince(i,j,k) \geq 0,\ \forall\ i = 1,\ldots,s,\ j = 1,\ldots,r,\ k = 1,\ldots,p$ and $\sum_{i=1}^{s}\sum_{j=1}^{r}\sum_{k=1}^{p} ince(i,j,k) = 1$, what means that total energy contained in the whole system is constant, equal to 1.

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

$$\Lambda = \left\{ ince \in [0,1]^{s\cdot r\cdot p} : \sum_{i=1}^{s}\sum_{j=1}^{r}\sum_{k=1}^{p} ince(i,j,k) = 1 \right.$$
$$\text{and } \forall\, i = 1,\ldots,s \ \sum_{j=1}^{r}\sum_{k=1}^{p} [ince(i,j,k) > 0] \leq q_i$$
$$\left. \text{and } \forall j = 1,\ldots,r,\ k = 1,\ldots,p \ \sum_{i=1}^{s} [ince(i,j,k) > 0] \leq 1 \right\} \tag{2}$$

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

EMAS space of states consists of the set of three-dimensional incedence and energy matrices:

$$X = \Lambda \tag{3}$$

Depending on context, it will be convenient to describe the state of the system by both $x \in X$ or $ince \in \Lambda$. The advantage of this solution may be clearly seen in Section 3.

2.3. EMAS behavior

Every agent starts its work in EMAS immediately after being activated. In every observable moment a certain agent gains the possibility of changing the state of the system by executing its action.

There are two types of agents' actions of agents may be divided into two distinct types:

- global – they change the state of the system in two or more locations, 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.

Functions $locsel$ and $agsel_{i \in Loc}$ (see (1)) are used do determine, which agent will be the next one to interact with the system. After being chosen, the agent chooses one of the possible actions, then it checks whether the associated condition is true, if so, the agent performs the action. The agent suspends its work in the system after performing the action which results in its death.

The more detailed description of computational agent behavior will be given by the algorithm pseudo-code (see Listing 1) in Section 2.5.

Every agent may perform actions contained in a predefined, finite set Act. The action, whose decision is to be evaluated by an agent, is chosen using the following function

$$\omega : U \times X \to \mathcal{M}(Act). \tag{4}$$

Notice, that the selection of action by all agents containing the same genotype gen in the same state x is performed according to the same probability distribution $\omega(gen, x)$ and does not depend on the whole agent's identifier $gen, n \in Ag$. In the simplest case ω returns the uniform probability distribution over Act for all $(gen, x) \in U \times X$

Every action $\alpha \in Act$ is the pair $(\delta_\alpha, (\{\vartheta_\alpha^{gen,n}\}, (gen, n) \in U \times \{1, \ldots, p\}))$ where

$$\delta_\alpha : U \times \{1, \ldots, p\} \times X \to \mathcal{M}(\{0, 1\}) \tag{5}$$

will denote the decision. The action α is performed with the probability $\delta_\alpha(gen, n, x)(\{1\})$ by the agent $ag_{gen,n}$ in the state $x \in X$ i.e. when the decision δ_α is undertaken (δ_α is positively evaluated). Moreover

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

defines the non-deterministic state transition caused by the execution of the action α by the agent $ag_{gen,n}$. The trivial state transition

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

such that for all A being the measurable set in X and all $x \in X$

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

is performed with the probability $\delta_\alpha(gen, n, x)(\{0\})$ i.e. when the decision δ_α is not undertaken (δ_α is evaluated as zero).

We assume that agents in EMAS perform the actions contained in the following set:

$$Act \quad = \quad \{repr, get, migr\} \tag{9}$$

where:

$repr$ the decision of get is based on checking whether there are neighboring agents in the same location, if certain neighbor has enough energy to reproduce, new agent is activated (its genotype is based on the genotype of its parents) and energy is transferred from the parents to the offspring agent,

get the decision of get is based on checking whether there are neighboring agents in the same location, if this condition is true, the agent chooses one of its neighbors and compares its genotype with own using predefined fitness function, then part of energy is transferred from worse to better agent,

$migr$ the decision of $migr$ is based on checking whether there are any neighboring locations, if this condition is true the agent is removed from its location and placed in one of neighboring locations.

The detailed descriptions of decisions and state transition functions for these actions may be found in [7].

We will use the family of functions $\varrho_\alpha^{gen,n} : X \to \mathcal{M}(X)$ where $gen \in U$, $n = 1, \ldots, p$, $\alpha \in Act$. Each of them expresses the probability transition imposed by agent $ag_{gen,n}$ that performs the action $\alpha \in Act$. They are given by the general formula:

$$\varrho_\alpha^{gen,n}(x)(A) = \delta_\alpha(gen, n, x)(\{0\}) \cdot \vartheta_{null}(x)(A) + \delta_\alpha(gen, n, x)(\{1\}) \cdot \vartheta_\alpha^{gen,n}(x)(A) \tag{10}$$

The definitions of the probability transition functions for each action is presented in observations 1–3 in [7].

2.4. Commutativity of local actions

Regarding the notions of global and local actions mentioned in Section 2.3 we divide the Act set in the following way:

$$Act = Act_{gl} \cup Act_{loc} \tag{11}$$

Speaking informally, local actions (elements of Act_{loc}) change only the entries of the layer $ince(i)$ of the incedence and energy matrix if the location $i \in Loc$ contains the agent performing certain action. Moreover these actions do not depend on other layers of $ince$. Action $null$ is obviously 'the most local one', because it does not change anything at all.

The above definition can be formalized as follows. The action $\alpha \in Act$ is *local* ($\alpha \in Act_{loc}$) when

1. α does not change anything except the part of the state (i.e. $ince(l)$ being the $l-$th layer in $ince$ matrix) that describes the location l in which $ag_{gen,n}$ is performing the action α:

$$\forall\, ince \in X : \varrho_{\alpha}^{gen,n}(ince)(A_l(ince)) = 1, \tag{12}$$

where

$$A_l(ince) = \{ince_{next} \in X : ince(i) = ince_{next}(i) \text{ for } i \neq l\},$$

and $ince_{next}$ denotes one of the state that can be reached at step immediately following the state in which the state x appears,

2. α is independent upon any other layers of $ince$:

$$\begin{gathered}\forall\, ince_1, ince_2 \in X, ince_1(l) = ince_2(l)\\ \forall B_1 \subset A_l(ince_1), B_2 \subset A_l(ince_2), \pi_l(B_1) = \pi_l(B_2) :\\ \varrho_{\alpha}^{gen,n}(ince_1)(B_1) = \varrho_{\alpha}^{gen,n}(ince_2)(B_2);\end{gathered} \tag{13}$$

here $\pi_l : [0,1]^{s\cdot r\cdot p} \to [0,1]^{r\cdot p}$ denotes the natural projection onto the l-th layer.

$$\pi_l(ince) = ince(l). \tag{14}$$

All other actions are considered global (elements of Act_{gl}).

Proposition 2.1. Let $\varrho_1, \varrho_2 : X \to \mathcal{M}(X)$ satisfy (12) and (13) with, respectively, l_1 and l_2, $l_1 \neq l_2$. Then

$$\int_X \varrho_2(y)(A)\varrho_1(x)(dy) = \int_X \varrho_1(y)(A)\varrho_2(x)(dy) \tag{15}$$

for all Borel measurable subsets A of X.

Proof Let us fix $x = ince \in X$. In the sequel we shall use a concise notation of l-th layer of x, i.e.

$$x_l = ince(l) \in [0,1]^{rp}.$$

Denote by $\varrho_{2,1}(x)(A)$ the left-hand side of (15). First let us show that the measure $\varrho_{2,1}(x)$ is concentrated on the following set.

$$A_{l_1,l_2}(x) = \{x' \in X : x'_i = x_i \text{ for } i \neq l_1 \text{ and } i \neq l_2\}$$

In other words, we shall show that

$$\varrho_{2,1}(x)(A_{l_1,l_2}(x)) = 1. \tag{16}$$

To this end, let us note that the following obvious inclusions hold.

$$A_{l_1}(x) \subset A_{l_1,l_2}(x), \quad A_{l_2}(x) \subset A_{l_1,l_2}(x)$$

Moreover, if we take any $y \in A_{l_1}(x)$ than for every $z \in A_{l_2}(y)$ we have

$$z_i = y_i \quad \text{for} \quad i \neq l_2$$

and, since $y_i = x_i$ for $i \neq l_1$, we have also that

$$z_i = x_i \quad \text{for} \quad i \neq l_1,\ i \neq l_2,$$

which means that

$$A_{l_2}(y) \subset A_{l_1,l_2}(x).$$

A consequence of the latter inclusion together with (12) is the following equality holding for all $y \in A_{l_1}(x)$.

$$\varrho_2(y)(A_{l_1,l_2}(x)) = 1. \tag{17}$$

Now note that another consequence of (12) is that

$$\varrho_{2,1}(x)(A_{l_1,l_2}(x)) = \int_X \varrho_2(y)(A_{l_1,l_2}(x))\varrho_1(x)(dy) = \int_{A_{l_1}(x)} \varrho_2(y)(A_{l_1,l_2}(x))\varrho_1(x)(dy).$$

Finally, from (17) and, again, (12) we have

$$\varrho_{2,1}(x)(A_{l_1,l_2}(x)) = \int_{A_{l_1}(x)} \varrho_1(x)(dy) = \varrho_1(x)(A_{l_1}(x)) = 1.$$

Hence (16) holds. Note that it is symmetric with respect to indices 1 and 2, so the same condition holds for $\varrho_{1,2}(x)$.

From (16) it follows that it suffices to prove (15) for $A \subset A_{l_1,l_2}(x)$. First consider a special type of such sets (sometimes called 'measurable rectangles'), i.e.

$$A = \{x' \in X : x'_{l_1} \in C_1,\ x'_{l_2} \in C_2,\ x'_i = x_i \text{ for other } i\} \tag{18}$$

with some Borel measurable $C_j \in [0,1]^{rp}$. As above, from (12) we know that

$$\int_X \varrho_2(y)(A)\varrho_1(x)(dy) = \int_{A_{l_1}(x)} \varrho_2(y)(A)\varrho_1(x)(dy).$$

Take $y \in A_{l_1}(x)$. There are two possible situations: $y \in A$ and $y \notin A$.
When $y \in A_{l_1}(x) \backslash A$, we have $y_i = x_i$ for $i \neq l_1$, so in particular $y_{l_2} \in C_2$ and therefore $y_{l_1} \notin C_1$. Moreover, for all $z \in A_{l_2}(y)$ we have $z_{l_1} = y_{l_1} \notin C_1$, which implies that $z \notin A$. In other words

$$A \cap A_{l_2}(y) = \emptyset.$$

On the other hand, when $y \in A_{l_1}(x) \cap A$, it is easy to see that

$$\pi_{l_2}(A \cap A_{l_2}(y)) = C_2 = \pi_{l_2}(A \cap A_{l_2}(x)).$$

Hence, from (13) it follows that

$$\varrho_2(y)(A \cap A_{l_2}(y)) = \varrho_2(x)(A \cap A_{l_2}(x)).$$

Since

$$\varrho_2(y)(A) = \varrho_2(y)(A \cap A_{l_2}(y)),$$

which is another consequence of (12), we have proven that

$$\varrho_2(y)(A) = \begin{cases} \varrho_2(x)(A \cap A_{l_2}(x)) & \text{if } y \in A_{l_1}(x) \cap A, \\ 0 & \text{if } y \in A_{l_1}(x) \backslash A. \end{cases} \tag{19}$$

Now let us compute the left-hand side of (15) for the above-defined A. From previous considerations it follows that

$$\varrho_{2,1}(x)(A) = \int_{A_{l_1}(x)} \varrho_2(y)(A \cap A_{l_2}(y)) \varrho_1(x)(dy).$$

Applying (19) we obtain

$$\begin{aligned} \varrho_{2,1}(x)(A) &= \int_{A \cap A_{l_1}(x)} \varrho_2(x)(A \cap A_{l_2}(x)) \varrho_1(x)(dy) = \varrho_2(x)(A \cap A_{l_2}(x)) \cdot \varrho_1(x)(A \cap A_{l_1}(x)) \\ &= \varrho_2(x)(A) \varrho_1(x)(A) \end{aligned}$$

As this equality is symmetric with respect to the layer number, we can repeat the same reasoning obtaining exactly the same result for $\varrho_{1,2}(x)(A)$, which means that

$$\varrho_{2,1}(x)(A) = \varrho_{1,2}(x)(A)$$

holds for 'rectangular' A. To finish the proof note that such 'rectangles' generate the σ-algebra of Borel measurable subsets of $A_{l_1,l_2}(x)$, thus the latter equality can be extended to the whole σ-algebra using standard properties of a measure. □

A straightforward consequence of the above proposition is the following corollary.

Corollary 2.1. Any two local actions executed by agents present in different locations commute.

Proof Take $gen, gen' \in U$, $n, n' \in \{1, \ldots, p\}$. Let $ag_{gen,n}$ perform a local action α in layer l and $ag'_{gen',n'}$ perform a local action α' in a layer l' different from l. The transition function of the composite action in which α is performed before α' is given by the following formula.

$$(\varrho_{\alpha'}^{gen',n'} \circ \varrho_{\alpha}^{gen,n})(x)(A) = \int_X \varrho_{\alpha'}^{gen',n'}(y)(A) \; \varrho_{\alpha}^{gen,n}(x)(dy) \tag{20}$$

Since both actions are local (i.e. they satisfy (12) and (13)) and they are performed in different locations, the assumptions of Proposition 2.1 are satisfied. From the proposition we obtain

$$(\varrho_{\alpha'}^{gen',n'} \circ \varrho_{\alpha}^{gen,n})(x)(A) = (\varrho_{\alpha}^{gen,n} \circ \varrho_{\alpha'}^{gen',n'})(x)(A) \tag{21}$$

□

Local actions must be mutually exclusive in the location and global actions are mutually exclusive in the whole system, so only one global action may be performed in one time moment in the system, but many local actions (at most one in each location) may be performed at the same time.

Observation 1. In the specific case of $EMAS$ global and local actions are following $Act_{gl} = \{migr\}$, $Act_{loc} = \{get, repr\}$.

Proof Taking into account Observations 2 and 4 in [7], the value of probability transition function imposed by get and $repr$ actions, performed by the agent $ag_{j,k}$ being present in the particular location l, depend only on the elements of the system state contained in its location (the layer $ince(l)$). Both actions may introduce changes only in entries of state in the same layer $ince(l)$. Hence actions get and $repr$ satisfy both locality conditions (12) and (13).

Action $migr$ does not satisfy conditions (12) and (13) because the value of the probability transition function imposed by this action depends not only on the elements of the system state contained in one location. The action $migr$ introduces changes in entries of state associated with the location different from the current location of migrating agent as well (see Observation 6 in [7]). □

2.5. Parallel execution of local actions in EMAS

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

Below we present the algorithms of computational agents, $LA_i, i \in Loc$ and MA using pseudocode inspired by C++ programming language.

Computational agent $CA = ag_{gen,n}$, present in the location i in every observable time moment chooses the action it wants to perform (using ω probability distribution to choose it randomly) and ask its supervisor – local agent LA_i for the permission, sending the communicate containing the chosen action with $send()$ function. Then it suspends its work waiting for the permission (or denial) that will be returned from LA_i using b_receive() blocking function. When the permission is granted and the decision assigned to the considered action is true, computational agent changes the state of the location (see Listing 1). Afterwards agent suspends its work again in order to get the permission for trying to perform subsequent action.

Here and later $\underline{A}(B)$ denotes the effect of random sampling one of the elements from the set B with random distribution A. We also assume that the sets $localact, globalact \subset Act$ contain the local and global actions' signatures respectively.

Local agent (see Listing 2) starts with checking whether the location contains any agents, in this case it sends the communicate to master agent, waits for the reply and then does nothing. Otherwise, when there are agents in the location, LA_i receives the signals containing actions to be performed from all its agents and puts them into hashmap indexed by genotypes and containing actions. Then the local agent utilizes $agsel_i$ function to choose the agent which should try to perform its action. This action is reported to master agent and after receiving permission, agent is granted the possibility to perform the action. All other agents (and the chosen one when the permission is not granted) are revoked to perform their actions. Afterwards agent waits for all local agents to report that they are ready to perform subsequent actions, report this fact to master agent and after receiving permission, lets them do it.

Master agent (see Listing 3) waits for all requests from location and then chooses randomly one location. If this location asked for the permission to perform global action, it is granted this permission and all other locations are revoked. Otherwise all locations that asked for the permission to perform global action are revoked and all asking for permission to perform local action – are granted. In the end,

master agent waits once more for all locations to report finishing their work and to let them try to perform subsequent action.

Listing 1. Computational agent's algorithm

```
while(1) {
    reply=0;
    α = ω(ag_{gen,n}, x);
    send(LA_i, α);
    b_receive(LA_i, rep);
    if(rep && δ_α(ag_{gen,n}, x) )
    {
        x_next = ϑ_α^{gen,n}(x)
    }
    send(LA_i);
    b_receive(LA_i);
}
```

2.6. Parallel EMAS dynamics

In the current section we present the parallel model of EMAS, that partially relaxes mutually exclusion among all actions (see [7]) allowing for performing local actions (performed in locations) in parallel, using introduced agents LA_{loc}, $loc \in Loc$ and MA as means for synchronization.

In the observable time moment all agents in all locations notify their local agents about their intent to perform an action, all local agents choose an agent with the distribution given by the $agsel_{loc}$, $loc \in Loc$ function and then notify master agent that they want to let to perform an action to one of their agents. Master agent chooses the location with the distribution given by $locsel$ function.

The probability that in the chosen location $i \in Loc$ agent wants to perform local action is as follows:

$$\xi_i(x) = \sum_{gen \in U} \sum_{n=1}^{p} (agsel_i(x)(\{gen, n\}) \cdot \omega(gen, x)(Act_{loc})) \tag{22}$$

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

$$\zeta^{loc}(x) = \sum_{i \in Loc} locsel(x)(\{i\})\xi_i(x) \tag{23}$$

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

$$(1 - \zeta^{loc}(x)) = \zeta^{gl}(x) \tag{24}$$

Listing 2. Local agent's algorithm

```
while(1) {
    localgen={U × 1,...,p ∋ (j,k) : ince(i,j,k) > 0}
    genact=hashmap(U × 1,...,p, Act);
    act=0;
    reply=0;
    if(#localgen==0)
    {
        send(MA,ϑ_null);
        b_receive(MA);
        send(MA,ϑ_null);
        b_receive(MA);
    }
    else
    {
        for(g ∈ localgen)
        {
          b_receive(g,act);
          genact[g]=act;
        }
        gchosen = agsel_i(x);
        report(genact[gchosen],gchosen);
    }
}

void report(act,chosen)
{
    send(MA, act);
    b_receive(MA,reply);
    if(reply)
        send(chosen,1);
    else
        send(chosen,0);
    for(g ∈ localgen \ {chosen} )
    {
        send(g,0);
    }
    for(g ∈ localgen)
    {
        b_receive(g);
    }
    send(MA);
    b_receive(MA);
    for(g ∈ localgen)
    {
        send(g);
    }
}
```

Listing 3. Master agent's algorithm

```
while(1) {
    local= {i : i ∈ [1, s]};
    localloc=∅;
    localglob=∅;
    act=0;
    rep=0;

    for(j ∈ local)
    {
        b_receive(j,act);
        if(act ∈ Act_global )
            localglob = localglob ∪ {j}
        else
            localloc = localloc ∪ {j}
    }
    lchosen = locsel(x);
    if(lchosen ∈ localglob)
    {
        send(lchosen,1);
        for(j in local \ {lchosen})
        {
            send(j,0);
        }
    }
    else
    {
        for(j ∈ localloc)
        {
            send(j,1);
        }
        for(j ∈ localglob)
        {
            send(j,0);
        }
    }
    for(j ∈ local)
    {
        b_receive(j);
    }
    for(j ∈ local)
    {
        send(j);
    }
}
```

If the global action was chosen, the state transition is as follows:

$$\tau^{gl}(x)(A) = \sum_{i \in Loc} locsel(x)(\{i\}) \cdot \left(\sum_{gen \in U} \sum_{n=1}^{p} agsel(x)(\{gen, n\}) \cdot \left(\sum_{\alpha \in Act_{gl}} \omega(gen, x)(\{\alpha\}) \cdot \varrho_{\alpha}^{gen,n}(x)(A) \right) \right) \tag{25}$$

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

$$Act_{+1loc} = \left\{ (\alpha_1, \ldots, \alpha_s) \in Act^s; \sum_{i=1}^{s} [\alpha_i \in Act_{loc}] > 0 \right\} \tag{26}$$

The probability that in i-th location the agent ag_{gen_i,n_i} chooses the action α_i is as follows:

$$\mu_{\alpha_i,gen_i,n_i}(x) = agsel_i(x)(\{gen_i, n_i\})\omega(gen_i, x)(\{\alpha_i\}) \tag{27}$$

Let us define the multi-index

$$ind = (\alpha_1, \ldots, \alpha_s; (gen_1, n_1), \ldots, (gen_s, n_s)) \in IND = Act^s \times (U \times \{1, \ldots, p\})^s \tag{28}$$

the probability that in consecutive locations agents ag_{gen_i,n_i} choose the actions α_i is given by:

$$\mu_{ind}(x) = \prod_{i=1}^{s} \mu_{\alpha_i,gen_i,n_i}(x) \tag{29}$$

Transition function for parallel system is following:

$$\tau^{loc}(x)(A) = \sum_{(\alpha_1,\ldots,\alpha_s) \in Act_{+1loc}} \sum_{ind \in IND} \mu_{ind}(x)(\pi_1^{ind}(x) \circ, \ldots, \circ \pi_s^{ind}(x))(A) \tag{30}$$

where:

$$\pi_i^{ind}(x) = \begin{cases} \varrho_{\alpha_i}^{gen_i,n_i}(x), & \alpha_i \in Act_{loc} \\ \vartheta_{null}, & \alpha_i \in Act_{gl} \end{cases} \tag{31}$$

Let us observe, that the value of $(\pi_1^{ind}(x), \circ, \ldots \circ \pi_s^{ind}(x))(A)$ does not depend on the composition order, because transition functions associates with local actions commutate pairwise (see Corollary 2.1 and Observation 1). It validates the following observation.

Observation 2. The probability transition function for parallel EMAS model is given by the formula

$$\tau(x)(A) = \zeta^{gl}(x)\tau^{gl}(x)(A) + \zeta^{loc}(x)\tau^{loc}(x)(A) \tag{32}$$

and the formulas (22) – (31).

Observation 3. The stochastic state transition of EMAS given by formula (32) satisfies the Markov condition.

Proof All transition functions and probability distributions given by formulas (22)–(31) depend only on the current state of the system, what motivates the Markovian features of the transition function τ given by (32). □

3. iEMAS extension

3.1. iEMAS structure

The *Immunologically based Evolutionary Multi-Agent System* (iEMAS) contains a dynamic collections of agents that belong to the predefined finite set Ag identical as in case of EMAS and 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 the similar structure as the agents previously defined, however their actions differ (because their goals differ from agents' goals) and their total energy does not have to be constant.

The iEMAS may be modeled as the following tuple

$$\begin{aligned}&\langle U, Loc, Top, Ag, Tc, \{typesel_i\}_{i\in Loc}, \{agsel_i\}_{i\in Loc},\\ &\{tcsel_i\}_{i\in Loc}, locsel, \{LA_i\}_{i\in Loc}, MA, \omega, \varphi, Act, Tcact\rangle\end{aligned} \tag{33}$$

where:

$typesel_i$ is the 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 the lymphocyte in i-th location to interact with the system in the current step,

φ is the decision function for the lymphocytes,

$Tcact$ is the set of actions that may be performed by the lymphocytes.

other symbols are defined similarly as in Section 2 (however some of them will be redefined in the course of this section).

The lymphocyte $Tc \ni tc_{gen}$ contains the genotype $gen \in U$ that plays the role of pattern that is used to detect similar genotypes by the means of predefined matching function $MATCH : U \times U \to \{0, 1\}$. It also posses its own energy that belongs to the interval $[0, tce_{max}]$.

3.2. iEMAS state

In addition to the EMAS state describing the location and energy of the agents (see (3)) we need to consider the 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 incedence and energy matrices in the following way:

$$\begin{aligned}\Gamma = \Bigg\{ &tcince \in [0, tce_{max}]^{r\cdot s} : \forall\, i = 1, \ldots, s \;\; \sum_{j=1}^{r} [tcince(i, j) > 0] \leq tcq_j \\ &\text{and } \forall\, j = 1, \ldots, r \;\; \sum_{i=1}^{s} [tcince(i, j) > 0] \leq 1 \Bigg\}\end{aligned} \tag{34}$$

where $tcince(i, j)$ stands for the 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 the particular locations. It is most convenient to assume $tcq_j = q_j$, $\forall j = 1, \ldots, s$.

The space of iEMAS state is given by:

$$X = \Lambda \times \Gamma \tag{35}$$

where Λ is given by (3).

3.3. iEMAS behavior

We introduce two disjoint action sets, first one is designed for agents

$$Act = \{repr, get, migr\} \tag{36}$$

The outcome of actions $repr$, $migr$ belonging to Act set is similar like in EMAS.In the case of iEMAS the domain of the decision and transition functions are formally extended because additional incidence and energy matrix for lymphocytes $tcince$ is added (see (35)). The decisions and the transition functions of these actions do not depend on and do not change the new state components in full extent. The effects of their operations are visible only on the part of the system state identical to EMAS. In particular, the changes of the state described before showing current state $x = ince$ and next state after the action $x_{next} = ince_{next}$, now have the form $x = (ince, tcince)$, $x_{next} = (ince_{next}, tcince_{next})$ respectively, where the index $next$ has similar meaning as in Section 2.4.

We also extend the trivial state transition ϑ_{null} in the same way.

The single action get inherited from EMAS is modified. When the agent performs this action, and its energy (or energy of evaluated agent) reaches zero, it activates the lymphocyte containing the genotype of inactivated agent.

The set of action designed for lymphocytes is

$$Tcact = \{give, kill\} \tag{37}$$

where:

$give$ is the action of the lymphocyte used to decrease its own energy (performed during every activation), which in the end causes the lymphocyte to be deactivated (when its energy reaches zero),

$kill$ is the action of removing the computational agent by a lymphocyte (performed when the genotype of the tested agent matches the pattern contained in the lymphocyte).

The detailed descriptions of decisions and state transition functions for these actions may be found in [7].

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

$$typesel_i : X \to \mathcal{M}\{0, 1\} \tag{38}$$

when 0 is chosen, one of agents will be activated, when 1 – lymphocyte.

The function choosing which agent will be activated $agsel_i$ is similar like in EMAS (see Section 2.1), but it now depends in some way on the extended state from X defined by (35). Now we introduce new function that will choose which lymphocyte will be activated:

$$tcsel_i : X \to \mathcal{M}(Tc) \tag{39}$$

Similarly as in case of $agsel_i$ the probability $tcsel_i(x)(\{gen\})$ vanishes when the lymphocyte tc_{gen} is inactive in the state $x \in X$.

The function ω choosing the action for the active agent remains intact, though its domain changes (because of the new state definition, see (35)).

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

$$\varphi : U \times X \to \mathcal{M}(Tcact) \tag{40}$$

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

$$\eta_\alpha^{gen}(x)(A) = \gamma_\alpha(gen, x)(\{0\}) \cdot \vartheta_{null}(x)(A) + \gamma_\alpha(gen, x)(\{1\}) \cdot \kappa_\alpha^{gen,n}(x)(A) \tag{41}$$

The definitions of the decisions γ_α and probability transition functions $\kappa_\alpha^{gen,n}$ for each action are given by Observations 8–9 in [7].

3.4. Parallel execution of local actions in iEMAS

We extend the formalism described in section 2.5 to describe the case of parallel iEMAS. We have divided the the set of EMAS actions into two disjoint sets local and global ones (see (11)). Because of the nature of actions performed by lymphocytes, all of them are considered as local.

Following coherency conditions (12), (13) introduced for EMAS we need to extend them for the case of iEMAS in the following way.

The action $\alpha \in Act$ and $\beta \in Tcact$ are local, when:

1. $\alpha \in Act$ and $\beta \in Tcact$ do not change anything except the part of the state ($l-$th layer in $ince$ matrix and l-th layer of $tcince$ matrix) that describes the location l in which $ag_{gen,n}$ is performing the action α, i.e.:

$$\forall x = (ince, tcince) \in X : \varrho_\alpha^{gen,n}(x)(\bar{A}_l(x)) = 1, \eta_\beta^{gen,n}(x)(\bar{A}_l(x)) = 1 \tag{42}$$

   where

$$\begin{gathered}\bar{A}_l(x) = \{(ince_{next}, tcince_{next}) \in X : \\ ince(i) = ince_{next}(i),\ tcince(i) = tcince_{next}(i) \text{ for } i \neq l\};\end{gathered} \tag{43}$$

2. $\alpha \in Act$ and $\beta \in Tcact$ are not dependent on any other layers of $ince$ and $tcince$:

$$\begin{gathered}\forall x_1, x_2 \in X, x_1(l) = x_2(l) \\ \forall B_1 \subset \bar{A}_l(x_1), B_2 \subset \bar{A}_l(x_2), \pi_l(B_1) = \pi_l(B_2) \\ \varrho_\alpha^{gen,n}(x_1)(B_1) = \varrho_\alpha^{gen,n}(x_2)(B_2), \\ \eta_\beta^{gen,n}(x_1)(B_1) = \eta_\beta^{gen,n}(x_2)(B_2)\end{gathered} \tag{44}$$

   where $x(l) = ((ince(l), tcince(l))$ and $tcince(l)$ denotes the l-th column of $tcince$. Moreover $\pi_l : [0,1]^{s \cdot r \cdot p} \times [0,1]^{s \cdot r} \to [0,1]^{r \cdot p} \times [0,1]^r$ denotes the natural projection on the l-th layer, i.e.

$$\pi_l(ince, tcince) = (ince(l), tcince(l)).$$

All other actions are considered global (elements of Act_{gl}).

Corollary 3.1. Any two local actions executed by agents or lymphocytes being present in different locations commute.

Proof The transition function of the composite of both actions may be also presented by the formulas similar to (20). Since both actions are local and executed in separate locations the condition analogous to Proposition 2.1 holds, for the measures defined now on the space $X = \Lambda \times \Gamma$ (mainly because of the same nature of sets generating the σ-algebra of Borel measurable sets on X). Hence desired commutativity holds. □

Observation 4. In the specific case of $iEMAS$ the set of local actions is composed of $Tcact \cup \{repr, get\}$ and set of global actions is $\{\text{migr}\}$.

Proof Taking into account Observations 11, 13 and 15 from [7], the value of probability transition function imposed by get, $kill$ and $give$ actions, depend only on the elements of the system state contained in one location (the layer $ince(l)$ and the column $tcince(l)$). Both actions may introduce changes only in the same sets of entries $ince(l)$ and $tcince(l)$. Hence actions get, $kill$ and $give$ satisfy both conditions (42) and (44).

Observation 1 already shows, that action $repr$ is local with respect to the space of states $X = \Lambda$. The extension of the space of states to $X = \Lambda \times \Gamma$ does not affect this condition. Similarly, globality of the action $migr$ shown in the Observation 1 holds for $X = \Lambda \times \Gamma$. □

Below we present the algorithms of lymphocytes, LA_i and MA adapted to the case of $iEMAS$. The algorithms for computational agent (see Listing 1) remains intact, however we need to introduce complementary algorithm for lymphocytes (see Listing 4).

Listing 4. Lymphocyte's algorithm

```
while(1) {
    reply=0;
    α = φ(tc_gen, x);
    send(LA_i, α);
    b_receive(LA_i, rep);
    if(rep && κ_α(tc_gen, x) )
    {
        x_next = η_α^gen(x)
    }
    send(LA_i);
    b_receive(LA_i);
}
```

Listing 5. Local agent's algorithm for iEMAS

```
while(1) {
   localgen={U × {1,...,p} ∋ (j,k) : ince(i,j,k) > 0}
   localtc={U ∋ j : tcince(i,j) > 0}
   genact=hashmap(U × {1,...,p}, Act);
   tcact=hashmap(U, Act);
   act=0;
   reply=0;
   if(# {localgen ∪ localtc } ==0)
   {
      send(MA,ϑ_null);
      b_receive(MA);
      send(MA,ϑ_null);
      b_receive(MA);
   }
   else
   {
      for(g ∈ localgen)
      {
        b_receive(g,act);
        genact[g]=act;
      }
      for(g ∈ localtc)
      {
        b_receive(g,act);
        tcact[g]=act;
      }
      if( typesel(x) )
      {
        gchosen=tcsel_i(x);
        report(tcact[gchosen],gchosen);
      }
      else
      {
        gchosen = agsel_i(x);
        report(genact[gchosen],gchosen);
      }

   }
}

void report(act,chosen)
{
   send(MA, act);
   b_receive(MA,reply);
   if(reply)
      send(chosen,1);
   else
      send(chosen,0);
   for(g ∈ {localgen ∪ localtc} \ {chosen} )
   {
      send(g,0);
   }
   for(g ∈ {localgen ∪ localtc})
   {
      b_receive(g);
   }
   send(MA);
   b_receive(MA);
   for(g ∈ {localgen ∪ localtc})
   {
      send(g);
   }
}
```

Listing 6. Master agent's algorithm for iEMAS

```
while(1) {
    local= {i : i ∈ [1, s]};
    localloc=∅;
    localglob=∅;
    act=0;
    rep=0;

    for(j ∈ local)
    {
        b_receive(j,act);
        if(act ∈ Act_global)
            localglob = localglob ∪ {j}
        else
            localloc = localloc ∪ {j}
    }
    lchosen = locsel(x);
    if(lchosen ∈ localglob)
    {
        send(lchosen,1);
        for(j in local \ {lchosen})
        {
            send(j,0);
        }
    }
    else
    {
        for(j ∈ localloc)
        {
            send(j,1);
        }
        for(j ∈ localglob)
        {
            send(j,0);
        }
    }
    for(j ∈ local)
    {
        b_receive(j);
    }
    for(j ∈ local)
    {
        send(j);
    }
}
```

This algorithm uses probability distribution function φ to choose the lymphocyte which will have possibility to perform the next action. Then similarly to computational agent function κ is used to evaluate the decision and η to perform the state transition after getting confirmation from local agent.

Because the lymphocytes were introduced into the system, the algorithm of LA_i must be modified for all $i \in Loc$, in order to make possible choosing lymphocyte for performing the action on the location (see Listing 5).

We extend the responsibilities of local agent by allowing it to choose the agent either from computational agents set or lymphocytes. Which one should be chosen in the current step is determined using $typesel$ function. Then the action chosen by the agent is reported to master agent, which allows for executing global or local actions using computational agents' action set alone, because all lymphocytes' actions are considered local (see Listing 6).

3.5. Parallel iEMAS dynamics

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

$$\xi_i(x) = typesel(x)(\{0\}) \sum_{gen \in U} \sum_{n=1}^{p} (agsel_i(x)(\{gen, n\}) \cdot \omega(gen, x)(Act_{loc})) + typesel(x)(\{1\}) \tag{45}$$

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

$$\zeta^{loc}(x) = \sum_{i \in Loc} locsel(x)(\{i\}) \xi_i(x) \tag{46}$$

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

$$(1 - \zeta^{loc}(x)) = \zeta^{gl}(x) \tag{47}$$

If the global action was chosen, the state transition is of similar shape as (25):

$$\tau^{gl}(x)(A) = \sum_{i \in Loc} locsel(x)(\{i\}) \cdot \left(\sum_{gen \in U} \sum_{n=1}^{p} agsel(x)(\{gen, n\}) \cdot \left(\sum_{\alpha \in Act_{gl}} \omega(gen, x)(\{\alpha\}) \cdot \varrho_{\alpha}^{gen,n}(x)(A) \right) \right) \tag{48}$$

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

$$Act_{+1loc} = \left\{ (\alpha_1, \ldots, \alpha_s) \in (Act \cup Tcact)^s; \sum_{i=1}^{s} [\alpha_i \in (Act_{loc} \cup Tcact)] > 0 \right\} \tag{49}$$

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:

$$\mu_{\alpha_i, gen_i, n_i, \widetilde{gen_i}}(x) = typesel(x)(\{0\}) \cdot agsel_i(x)(\{gen_i, n_i\}) \omega(gen_i, x)(\{\alpha_i\}) + typesel(x)(\{1\}) tcsel_i(x)(\{\widetilde{gen_i}\}) \varphi(\widetilde{gen_i}, x)(\{\alpha_i\}) \tag{50}$$

Let us redefine the multi-index introduced in formula (28):

$$\begin{aligned} ind &= \big(\alpha_1,\ldots,\alpha_s;(gen_1,n_1),\ldots,(gen_s,n_s);(\widetilde{gen_1}),\ldots,(\widetilde{gen_s})\big) \\ &\in IND = (Act \cup Tcact)^s \times (U \times \{1,\ldots,p\})^s \times U^s \end{aligned} \tag{51}$$

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

$$\mu_{ind}(x) = \prod_{i=1}^{s} \mu_{\alpha_i,gen_i,n_i,\widetilde{gen_i}}(x). \tag{52}$$

Transition function for parallel system is following:

$$\tau^{loc}(x)(A) = \sum_{(\alpha_1,\ldots,\alpha_s)\in Act_{+1loc}} \sum_{ind\in IND} \mu_{ind}(x)(\pi_1^{ind}(x)\circ,\ldots,\circ\,\pi_s^{ind}(x))(A) \tag{53}$$

where π_i introduced in formula (31) is redefined as:

$$\pi_i^{ind}(x) = \begin{cases} \varrho_{\alpha_i}^{gen_i,n_i}(x), & \alpha_i \in Act_{loc} \\ \eta_{\alpha_i}^{\widetilde{gen_i}}(x), & \alpha_i \in Tcact \\ \vartheta_{null}, & \alpha_i \in Act_{gl} \end{cases} \tag{54}$$

Let us observe again, that the value of $(\pi_1^{ind}(x)\circ,\ldots,\circ\,\pi_s^{ind}(x))(A)$ does not depend on the composition order, because transition functions associates with local actions commutate pairwise. Finally, we may derive the following observation.

Observation 5. The probability transition function for parallel iEMAS model is given by the formula

$$\tau(x)(A) = \zeta^{gl}(x)\tau^{gl}(x)(A) + \zeta^{loc}(x)\tau^{loc}(x)(A) \tag{55}$$

and the formulas (45) – (54).

Observation 6. The stochastic state transition of iEMAS given by formula (55) satisfies the Markov condition.

Proof All transition functions and probability distributions given by formulas (45)–(54) depend only on the current state of the system, what motivates the Markovian features of the transition function τ given by (55). □

4. Conclusions

The paper extends the EMAS and iEMAS architecture defined in [7] to the desirable case in which some agent's action can be executed in parallel in the single step of evolution. The additional governing agents (managing agent and local agents) equipped with the mechanism of blocking communication among them are introduced. These agents allow for executing single global action or at most one local action in each location in the single evolution epoch. Again, as in [7] the stationary Markov chain models for

these systems are defined and verified. The space of states and the definitions of transition operators associated with all agent's and T-cell's actions described in [7] was utilized.

The main tool for obtaining the final formula is the Bayes theorem which allow to derive the probability transition function as the barycentric combination of elementary stochastic transformations implemented in particular actions. The crucial feature that allow to establish the probability transition function in case of parallel execution of local actions is commutativity of their transition operators. We formulate the abstract conditions of such commutativity which allow for easy classifying of agent's or T-cell's actions as local or global ones.

The obvious advantage of the EMAS and iEMAS architecture proposed in this paper over the previous versions is their better computational efficiency caused by the relaxed synchronization among actions.

The parallel iEMAS can be especially dedicated to the cases with costly fitness (see [3, 5]), which is performed as a long-lasting iterative process increasing the evaluation accuracy. The well-tuned, by immunological evolution, lymphocytes can early recognize and make the unpromising agents inactive on the basis of its rough fitness value, sparing a significant part of CPU time. All actions of T-cells are local ones, so their can be executed in parallel in separate locations.

However the general form of the Markov kernels associated wit the agent's actions were identified, we intend to identify their numerical and functional parameters as the probability distributions of mixing operator MIX, agent selection functions $agsel, \omega, typesel, tcsel, \varphi$ and energy thresholds of agents and lymphocytes.

The Markov model presented in this paper constitute the basis of the asymptotic analysis of EMAS ans iEMAS systems in order to evaluate their search possibilities and efficiency. It seems to be the way of verifying their probabilistic guarantee of success (e.g. by checking the ergodicity) and comparing with the other stochastic algorithms by comparing their limit invariant measures (if they exist).

References

[1] Back, T., Hammel, U., Schwefel, H.-P.: Evolutionary computation: Comments on the history and current state, *IEEE Trans. on Evolutionary Computation*, **1(1)**, 1997.

[2] Byrski, A., Kisiel-Dorohinicki, M.: Immunological selection mechanism in agent-based evolutionary computation, *Intelligent Information Processing and Web Mining : proceedings of the international IIS: IIPWM '05 conference : Gdansk, Poland* (M. A. Klopotek, S. T. Wierzchon, K. Trojanowski, Eds.), Advances in Soft Computing, Springer Verlag, 2005.

[3] Byrski, A., Kisiel-Dorohinicki, M.: Agent-Based Evolutionary and Immunological Optimization, *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, *Proc. of the IASTED Int. Symp.: Applied Informatics* (M. Hamza, Ed.), IASTED/ACTA Press, 2002.

[5] Byrski, A., Schaefer, R.: Immunological mechanism for asynchronous evolutionary computation boosting, *ICMAM 2008 : European workshop on Intelligent Computational Methods and Applied Mathematics : an international forum for researches, teachers and students : Cracow, Poland*, 2008.

[6] Byrski, A., Schaefer, R.: Formal Model for Agent-Based Asynchronous Evolutionary Computation, *IEEE Congress of Evolutionary Computation, Trondheim, Norway (submitted to)*, IEEE, 2009.

[7] Byrski, A., Schaefer, R.: Stochastic Model of Evolutionary and Immunological Multi-Agent Systems: Mutually Excluded Actions, *Fundamenta Informaticae*, 2009.

[8] Cantú-Paz, E.: A summary of research on parallel genetic algorithms, *IlliGAL Report No. 95007. University of Illinois*, 1995.

[9] Cetnarowicz, K., Kisiel-Dorohinicki, M., Nawarecki, E.: The application of evolution process in multi-agent world (MAW) to the prediction system, *Proc. of the 2nd Int. Conf. on Multi-Agent Systems (ICMAS'96)* (M. Tokoro, Ed.), AAAI Press, 1996.

[10] Jennings, N. R., Sycara, K., Wooldridge, M.: A Roadmap of Agent Research and Development, *Journal of Autonomous Agents and Multi-Agent Systems*, **1**(1), 1998, 7–38.

[11] Jennings, N. R., Wooldridge, M. J.: Software Agents, *IEE Review*, 1996, 17–20.

[12] Kisiel-Dorohinicki, M.: Agent-Oriented Model of Simulated Evolution, *SofSem 2002: Theory and Practice of Informatics* (W. I. Grosky, F. Plasil, Eds.), 2540, Springer-Verlag, 2002.

[13] Michalewicz, Z.: *Genetic Algorithms Plus Data Structures Equals Evolution Programs*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1994, ISBN 0387580905.