Asymptotic Analysis of Computational Multi-Agent Systems

Aleksander Byrski[1], Robert Schaefer[1,4], Maciej Smołka[2], Carlos Cotta[3,5]

[1]AGH University of Science and Technology, Kraków, Poland
[2]Jagiellonian University, Kraków, Poland
[3]University of Málaga, Málaga, Spain
{olekb,schaefer}@agh.edu.pl, smolka@ii.uj.edu.pl, ccottap@lcc.uma.es

Abstract. A stationary Markov chain model of the agent-based computation system EMAS is presented. The primary goal of the model is better understanding the behavior of this class of systems as well as their constraints. The ergodicity of this chain can be verified for the particular case of EMAS, thus implying an asymptotic guarantee of success (the ability of finding all solutions of the global optimization problem). The presented model may be further adapted to numerous evolutionary and memetic systems.

1 Motivation

Evolutionary algorithms (EAs) and multi-agent systems (MAS) are closely related paradigms. Among other similarities, they share conceptual elements such as the usage of a pool of entities (individuals in the case of EAs, agents in the case of MAS) which interact among themselves directly (via n-ary operators in EAs, and using autonomous, proactive or reactive behaviors in MAS) or indirectly (via modifications of the environment in MAS, and by e.g. coevolution, archive-based strategies, etc. in EAs). Not surprisingly, cross-fertilization of both paradigms has been attempted in the so-called agent-based computational systems (e.g. EMAS [3], AMAS [23], GCE [7]). In particular, EMAS (Evolutionary Multi-Agent System introduced by Cetnarowicz et al. in [6]) have been shown to be effective in solving difficult optimization tasks, e.g., optimization of neural-network architectures, multi-modal optimization, multi-criteria optimization.

The connection of MAS and EAs is particularly clear in the case of memetic algorithms (MA). It is customary – and in some sense based on good practical reasons – to consider that a MA is an EA hybridized with some form of local search (LS). This definition of MA was actually popularized by early works such as [18], and paved the way for the vigorous development of optimization algorithms based on this idea (exhibiting a remarkable record of success, check

e.g. [8]). It is also true that seminal works on this topic had a wider perspective, in which EAs endowed with LS was rather an appropriate incarnation of a MA than a restrictive definition [14]. Under this wider interpretation of the MA paradigm, a stronger relationship with multi-agent systems emerges. Indeed, a MA has been sometimes defined (as early as in [17] – see also [15]) as a cooperative-competitive strategy of optimizing agents. The use of the term *agent* here tries to emphasize the fact that individuals are more than mere solution placeholders that passively suffer the application of different variation and selection operations on them [16]. On the contrary they can be regarded as active actors in the search process, intertwining periods of individual search/learning with periods of cooperation and competition. While this interpretation remains compatible with classical MA approaches, it also opens up the door to more complex strategies, e.g. individual roles [1], different recombination behaviors [2], etc.

The relationship between multi-agent systems and memetic algorithms is not limited to a simple source of algorithmic inspiration. On the contrary, it can provide a useful means for improving the theoretical understanding of these techniques. In this sense, and opposed to classical evolutionary strategies for which qualitative formal models were introduced and intensively studied (see e.g. [24], [21], [20]), it must be noted that there is still lack of them for most complex, biologically-inspired heuristics.

Based on the results presented in [4, 22] we introduce a discrete, finite state-space Markov chain as a model for EMAS. We also present a theorem of ergodicity of such a system. This feature in not so obvious in this case as in case of simple genetic mechanisms, where the passage between two arbitrary states is possible in a single step, if the mutation rate is strictly positive [24].

Such an analysis may ensure that the system is able to reach a population containing an arbitrary minimizer in a finite number of steps. Moreover the effective upper bound of step number may be evaluated. In addition, asymptotic guarantee of success is satisfied [9, 19]. In the course of modelling a number of constraints were indicated, leading to better understanding of the functioning of these systems (e.g. synchronization schemes, probability distributions used and topology of connections).

2 EMAS architecture and behavior

We will focus on a EMAS systems solving global optimization problems consisting of finding all global minimizers $\arg\min\{FITN(x)\}, x \in U$ where $FITN : U \to \mathbb{R}_+$, and U is a finite genetic universum $\#U = r < +\infty$.

Computational EMAS agents belong to the predefined finite set Ag one-to-one mapped on set $U \times P$, where $P = \{1, \dots, p\}$ and p is assumed to be the maximum number of agents contain the same genotype, so each agent $ag_{gen,n} \in Ag$ is uniquely represented by its signature $(gen, n) \in U \times P$.

Agents are assigned to locations $Loc = \{1, \dots, s\}$. The locations are linked by 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. Each agent possesses a variable parameter called energy, its value is quantized and belongs to $\{0, \Delta e, 2 \cdot \Delta e, 3 \cdot \Delta e, \ldots, m \cdot \Delta e\}$. The current value of the energy exhibits the maturity of agent in solving the optimization problem, affecting its abilities (reproduction, cloning, migration) (see [10]).

Let us introduce the set of three-dimensional, incidence and energy matrices $x \in X$ with s layers (corresponding to all locations) $x(i) = \{x(i, gen, n), \ gen \in U, \ n \in P\}$, $i \in Loc$. The layer $x(i)$ will contain energies of agents in i-th location. In other words, $x(i, gen, k) > 0$ 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:

– 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 $\Delta e \leq \frac{1}{\sum_{i=1}^{s} q_i}$ what allows 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,

– $(\cdot, j, k)$-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,

– entries in the incidence and energy matrices 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 the total energy contained in the whole system is constant, equal to 1.

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

$$X = \Big\{ x \in \{0, \Delta e, 2 \cdot \Delta e, 3 \cdot \Delta e, \ldots, m \cdot \Delta e\}^{s \cdot r \cdot p}, \ \Delta e \cdot m = 1,$$

$$\sum_{i=1}^{s} \sum_{j=1}^{r} \sum_{k=1}^{p} x(i, j, k) = 1 \text{ and } \forall \, i = 1, \ldots, s \ \sum_{j=1}^{r} \sum_{k=1}^{p} [x(i, j, k) > 0] \leq q_i$$

$$\text{and } \forall j = 1, \ldots, r, \ k = 1, \ldots, p \ \sum_{i=1}^{s} [x(i, j, k) > 0] \leq 1 \Big\} \tag{1}$$

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

Note that the formula (1) implies that there must exist at least one agent in the system i.e. at least one location is non-empty at a time.

EMAS may be modeled as the following tuple:

$$< U, Loc, Top, Ag, \{agsel_i\}_{i \in Loc}, locsel, \{LA_i\}_{i \in Loc}, MA, \omega, Act > \quad (2)$$

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 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_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.

$agsel_i : X \to \mathcal{M}(U \times P)$ 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 $agsel_i$. The probability $agsel_i(x)(gen, n)$ vanishes when the agent $ag_{gen,n}$ is inactive in the state $x \in X$ or it is present in other than i-th location,

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

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

The population of agents is initialized by using introductory sampling. It may be explained as a one-time sampling from X according to the predefined probability distribution (possibly uniform) from $\mathcal{M}(X)$. Every agent starts its work in EMAS immediately after being activated. At every observable moment a certain agent on each location gains the possibility of changing the state of the system by executing its action.

The function $agsel_i$ is used by the Local Agent LA_i to determine which agent present on i-th location will be the next one to interact with the system. After being chosen, the agent $ag_{gen,n}$ chooses one of the possible actions according to the probability distribution $\omega(x, gen)$. Notice the relationship of this probability distribution with the concept of fine-grain schedulers introduced in the syntactic model for memetic algorithms in [12].

Next, the agent applies to LA_i for the permission to perform this action. When the permission is granted, $ag_{gen,n}$ checks whether the associated condition is true, and if so, the agent performs the action. The agent suspends its work in the system after performing the action which brings its energy to zero.

Master agent MA manages the activities of LA_i allowing them to grant permissions for their agents (thus relating to coarse-grain schedulers in [12]). Each action $\alpha \in Act$ is the pair of families of random functions $\{\delta_\alpha^{gen,n}\}_{gen \in U, n \in P}$ and $\{\vartheta_\alpha^{gen,n}\}_{gen \in U, n \in P}$ where

$$\delta_\alpha^{gen,n} : X \to \mathcal{M}(\{0, 1\}) \quad (3)$$

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

$$\vartheta_\alpha^{gen,n} : X \to \mathcal{M}(X) \tag{4}$$

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

$$\vartheta_{null} : X \to \mathcal{M}(X) \tag{5}$$

such that for all $x \in X$

$$\vartheta_{null}(x)(x') = \begin{cases} 1 & \text{if } x = x' \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

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

The value of the probability transition function for action α for the agent containing the n-th copy of genotype gen being in the location l

$$\varrho_\alpha^{gen,n} : X \to \mathcal{M}(X) \tag{7}$$

for the arbitrary current state $x \in X$ and the next one $x' \in X$ is given by:

$$\varrho_\alpha^{gen,n}(x)(x') = \delta_\alpha^{gen,n}(x)(0) \cdot \vartheta_{null}(x)(x') + \delta_\alpha^{gen,n}(x)(1) \cdot \vartheta_\alpha^{gen,n}(x)(x') \tag{8}$$

Notice finally that it is formally possible to consider a very large (yet finite) set *Act*, comprising all actions up to a certain description length (using a Gödel numbering or any appropriate encoding). This implies that this set may be implicitly defined by such an encoding, allowing much flexibility in the set of actions available (a connection can be drawn with multimeme algorithms [11]).

The agents' actions may belong to one of 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.

In the system governed by software agents there will be either a possibility of performing many local or one global action at a time.

3 EMAS dynamics

At the observable moment at which EMAS takes 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 using 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 using the probability distribution given by $locsel(x)$.

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

$$\xi_i(x) = \sum_{gen \in U} \sum_{n=1}^{p} agsel_i(x)(gen, n) \cdot \omega(x, gen)(Act_{loc}) \qquad (9)$$

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

$$\zeta^{loc}(x) = \sum_{i \in Loc} locsel(x)(i) \cdot \xi_i(x) \qquad (10)$$

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

$$\zeta^{gl}(x) = 1 - \zeta^{loc}(x) \qquad (11)$$

If a global action has been chosen, the state transition is as follows:

$$\tau^{gl}(x)(x') = \sum_{i \in Loc} locsel(x)(i)$$

$$\left(\sum_{gen \in U} \sum_{n=1}^{p} agsel_i(x)(gen, n) \cdot \left(\sum_{\alpha \in Act_{gl}} \omega(x, gen)(\alpha) \cdot \varrho_\alpha^{gen,n}(x)(x') \right) \right) \qquad (12)$$

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

$$Act_{+1loc} = \left\{ (\alpha_1, \dots, \alpha_s) \in Act^s; \ \sum_{i=1}^{s} [\alpha_i \in Act_{loc}] > 0 \right\} \qquad (13)$$

Let us define now the family of coefficients $\{\mu_{\alpha_i, gen_i, n_i}(x)\}$, $i \in Loc$, $gen_i \in U$, $n_i \in P$, $x \in X$. If the location i is nonempty at the state x, then $\mu_{\alpha_i, gen_i, n_i}(x)$ is equal to the probability that in the i-th location agent ag_{gen_i, n_i} chooses action α_i:

$$\mu_{\alpha_i, gen_i, n_i}(x) = agsel_i(x)(gen_i, n_i) \cdot \omega(x, gen_i)(\alpha_i). \qquad (14)$$

Of course $\mu_{\alpha_i, gen_i, n_i}(x) = 0$ if agent ag_{gen_i, n_i} does not exist in location i at state x, because $agsel_i(x)(gen_i, n_i) = 0$ in this case. Moreover, we set $\mu_{\alpha_i, gen_i, n_i}(x) = 1$ if location i is empty at state x. Next we introduce the multi-index:

$$ind = \big(\alpha_1, \dots, \alpha_s; (gen_1, n_1), \dots, (gen_s, n_s)\big) \in IND = Act_{+1loc}^s \times (U \times P)^s. \qquad (15)$$

The probability that at state x, in consecutive locations agents ag_{gen_i, n_i} choose actions α_i is given by:

$$\mu_{ind}(x) = \prod_{i=1}^{s} \mu_{\alpha_i, gen_i, n_i}(x) \qquad (16)$$

the transition function for the case of parallel executing of local actions is then:

$$\tau^{loc}(x)(x') = \sum_{ind \in IND} \mu_{ind}(x)(\pi_1^{ind} \circ, \ldots, \circ \pi_s^{ind})(x)(x') \qquad (17)$$

where

$$\pi_i^{ind}(x) = \begin{cases} \varrho_{\alpha_i}^{gen_i, n_i}(x), & \alpha_i \in Act_{loc} \text{ and the location } i \text{ is nonempty} \\ \vartheta_{null}, & \alpha_i \in Act_{gl} \text{ or the location } i \text{ is empty.} \end{cases} \qquad (18)$$

It is possible to prove that the value of $(\pi_1^{ind} \circ, \ldots, \circ \pi_s^{ind})(x)(x')$ does not depend on the composition order because transition functions associated with local actions commutate pairwise. The proof of this property in the discrete model is similar to the proof in [22] for a continuous system state space, and is omitted here due to space constraints.

The commutativity of local action validates the following observation:

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

$$\tau(x)(x') = \zeta^{gl}(x) \cdot \tau^{gl}(x)(x') + \zeta^{loc}(x) \cdot \tau^{loc}(x)(x') \qquad (19)$$

and formulas (9)–(18).

It is also easy to see that

Observation 2 *The stochastic state transition of EMAS given by formula (19) satisfies the Markov condition. Moreover, the Markov chain defined by these functions is stationary.*

4 Sample actions and asymptotic behavior

Let us consider a sample EMAS with the following set of actions:

$$Act = \{get, repr, clo, migr\} \qquad (20)$$

Due to space limitations we describe the actions informally, underlining only the necessary conditions for the subsequent analysis of the systems's ergodicity. Complete formal descriptions of these actions leading to the probability transition functions (3) and (4) may be found in [4]. In the following (gen, n) stands for the signature of a generic agent that attempts to execute the following actions:

get Decision $\delta_{get}^{gen,n}$ for energy transfer is positive when there is at least one agent more on the same location. Agent chooses randomly one of its neighbors and during the meeting, the energy is exchanged between agents, what may be considered somewhat as a tournament (see tournament selection [13]). The direction of the energy flow is determined by a probability distribution $CMP : U \times U \to \mathcal{M}(\{0, 1\})$ dependent on agents' fitnesses and the current state of the system. In the next state one of the agents receives a predefined part of energy Δe from its neighbor, which is assumed to satisfy $\Delta e \leq \left(\sum_{i=1}^s q_i\right)^{-1}$.

repr Decision $\delta_{repr}^{gen,n}$ for reproduction is positive when the energy of the agent performing the action is greater than a reproduction threshold e_{repr} and there is at least one agent more in the same location satisfying the same energy condition. We assume that $e_{repr} \leq 2\Delta e$. These agents create an offspring agent based on their solutions using a predefined mixing operator. Part of the parents' energy ($e_0 = n_0 \cdot \Delta e$, n_0 is even) is passed to the offspring.

clo Decision $\delta_{clo}^{gen,n}$ for cloning is based on checking the amount of agent's energy only. An agent with enough energy strictly greater than Δe, creates an offspring agent based on its solution (applying a predefined mutation operator $MUT : U \to \mathcal{M}(U)$). Part of the parent's energy Δe is passed to the offspring.

migr Decision $\delta_{migr}^{gen,n}$ is positive when an agent has enough energy greater than e_{migr} and there exists a location that is able to accept it (the number of agents there is lower than its capacity). When these conditions are met the agent is moved from its location to another. We assume, that $e_{migr} < s^{-1}$.

Theorem 1. *Given the following assumptions:*

1. *The capacity of every location is greater than one, $q_i > 1, i = 1, \ldots, s$.*
2. *The graph of locations is connected.*
3. *Each active agent can be selected by its local agent with strictly positive probability, so*
   $\exists \iota_{agsel} > 0; \ \forall \ i \in Loc, \forall \ gen \in U, \forall \ n \in P, \ \forall \ x \in \{y \in X; y(i, gen, n) > 0\},$
   $agsel_i(x)(gen, n) \geq \iota_{agsel}.$
4. *The families of probability distributions being the parameters of EMAS have the uniform, strictly positive lower bounds:*
   $\exists \ \iota_\omega > 0; \ \forall \ x \in X, \ gen \in U, \ \alpha \in Act, \ \omega(gen, x)(\alpha) \geq \iota_\omega,$
   $\exists \ \iota_{CMP} > 0; \ \forall \ gen, gen' \in U, \ CMP(gen, gen') \geq \iota_{CMP},$
   $\exists \ \iota_{mut} > 0; \forall gen, gen' \in U, \ MUT(gen)(gen') \geq \iota_{mut},$
   $\exists \ 0 < \iota_{locsel} < 1; \forall \ x \in X, \forall \ j \in Loc, \ locsel(x)(j) \geq \iota_{locsel}.$

We can construct a finite sequence of transitions between two arbitrarily chosen system states which may be passed with strictly positive probability. Moreover we can deliver the upper bound of the number of such transitions, which can be effectively computed based on the system's parameters.

The proof of the Theorem 1 is omitted in this paper because of its length and strictly technical substance. It has already been completed and will be published in an extended version.

Assumptions 1 and 2 allow to migrate agents to all locations that are not over-populated (with a positive probability). The positive probability of performing crucial actions (*get*, *clo*) changing energy and genotype is ensured by assumptions 3 and 4. The above stated properties make possible to define a generic path between two arbitrary states of the system.

Notice that verifying the ergodicity is different than usually done for classical genetic algorithms (see e.g. [24]), where all possible states of the system are reachable within a single step, because of the characteristics of the mutation operator.

Remark 1. Theorem 1 makes all states containing the extrema reachable in a finite number of states, thus EMAS satisfies an asymptotic guarantee of success [9], [19]. Moreover the Markov chain modelling EMAS (see equation (19)) is ergodic.

5   Conclusions

We presented a discrete version of EMAS model (following the continuous versions of the model published in [4, 22]). The space of states of the system – Eq. (1) – and the probability transition function – Eq. (19) – constitute a stationary Markov chain.

Under assumptions of Theorem 1 an EMAS is able to reach the population containing an arbitrary minimizer in a finite number of steps. The effective upper bound for the number of steps required may also be evaluated. In addition, asymptotic guarantee of success is satisfied [9, 19]. The properties mentioned above make the Markov chain modelling EMAS ergodic. The ergodicity in the case of EMAS is not as straightforward as in classical genetic algorithms (cf. the works of Vose [24]) where any possible state of the system may be reached in one step thanks to positive mutation rates.

In the course of modelling several constraints were indicated leading to better understanding of the functioning of agent-based memetic systems (e.g. synchronization schemes, probability distributions used and topology of connections).

References

1. R. Berretta, C. Cotta, and P. Moscato. Enhancing the performance of memetic algorithms by using a matching-based recombination algorithm: Results on the number partitioning problem. In M. Resende and J. Pinho de Sousa, editors, *Metaheuristics: Computer-Decision Making*, pages 65–90. Kluwer Academic Publishers, Boston MA, 2003.
2. R. Berretta and P. Moscato. The number partitioning problem: An open challenge for evolutionary computation ? In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 261–278. McGraw-Hill, 1999.
3. A. Byrski and M. Kisiel-Dorohinicki. Agent-based evolutionary and immunological optimization. In *Computational Science - ICCS 2007, 7th International Conference, Beijing, China, May 27 - 30, 2007, Proceedings*. Springer, 2007.
4. Aleksander Byrski and Robert Schaefer. Stochastic model of evolutionary and immunological multi-agent systems: Mutually exclusive actions. *Fundamenta Informaticae*, 95(2-3):263–285, 2009.
5. E. Cantú-Paz. *Efficient and accurate parallel genetic algorithms*, volume 1 of *Genetic Algorithms and Evolutionary Computation*. Springer-Verlag, 2000.
6. K. Cetnarowicz, M. Kisiel-Dorohinicki, and E. Nawarecki. The application of evolution process in multi-agent world (MAW) to the prediction system. In M. Tokoro, editor, *ICMAS'96 Proceedings*. AAAI Press, 1996.
7. C. Chira, A. Gog, and D. Dumitrescu. Exploring population geometry and multi-agent systems: a new approach to developing evolutionary techniques. In *Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computation*, pages 1953–1960, Atlanta, GA, 2008. ACM Press.

8. W.E. Hart, N. Krasnogor, and J.E. Smith. Memetic evolutionary algorithms. In *Recent advances in memetic algorithms*, volume 166 of *Studies in Fuzziness and Soft Computing*, pages 3–27. Springer-Verlag, 2005.

9. R. Horst and P. Pardalos. *Handbook of Global Optimization*. Kluwer, 1995.

10. M. Kisiel-Dorohinicki. Agent-oriented model of simulated evolution. In William I. Grosky and Frantisek Plasil, editors, *SofSem 2002: Theory and Practice of Informatics*, volume 2540 of *LNCS*. Springer-Verlag, 2002.

11. N. Krasnogor and S. Gustafson. A study on the use of "self-generation" in memetic algorithms. *Natural Computing*, 3:53–76, 2004.

12. N. Krasnogor and J. Smith. A tutorial for competent memetic algorithms: Model, taxonomy, and design issues. *IEEE Transactions on Evolutionary Computation*, 9(5):474–488, 2005.

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

14. P. Moscato. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. Technical Report Caltech Concurrent Computation Program, Report. 826, California Institute of Technology, Pasadena, California, USA, 1989.

15. P. Moscato. Memetic algorithms: A short introduction. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 219–234. McGraw-Hill, 1999.

16. P. Moscato and C. Cotta. A gentle introduction to memetic algorithms. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, pages 105–144. Kluwer Academic Publishers, Boston MA, 2003.

17. M.G. Norman and P. Moscato. A competitive and cooperative approach to complex combinatorial search. Technical Report Caltech Concurrent Computation Program, Report. 790, California Institute of Technology, Pasadena, California, USA, 1989.

18. N.J. Radcliffe and P.D. Surry. Formal Memetic Algorithms. In T. Fogarty, editor, *Evolutionary Computing: AISB Workshop*, volume 865 of *Lecture Notes in Computer Science*, pages 1–16. Springer-Verlag, Berlin, 1994.

19. A. Rinnoy Kan and G. Timmer. Stochastic global optimization methods. *Mathematical Programming*, 39:27–56, 1987.

20. G. Rudolph. Models of stochastic convergence. In T. Bäck, D.B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computations*. Oxford University Press, 1997.

21. G. Rudolph. Stochastic processes. In T. Bäck, D.B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computations*. Oxford University Press, 1997.

22. Robert Schaefer, Aleksander Byrski, and Maciej Smołka. Stochastic model of evolutionary and immunological multi-agent systems: Parallel execution of local actions. *Fundamenta Informaticae*, 95(2-3):325–348, 2009.

23. Abu S. S. M. Barkat Ullah, Ruhul Sarker, and Chris Lokan. An agent-based memetic algorithm (AMA) for nonlinear optimization with equality constraints. In *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, pages 70–77, Trondheim, Norway, 2009. IEEE Press.

24. M. Vose. *The Simple Genetic Algorithm: Foundations and Theory*. MIT Press, Cambridge, MA, USA, 1998.