A Patchwork Model for Evolutionary Algorithms with Structured and Variable Size Populations

Thiemo Krink
Department of Computer Science
Aarhus University
DK-8000 Aarhus C
Denmark
e-mail: *krink@daimi.au.dk*

Brian H. Mayoh
Department of Computer Science
Aarhus University
DK-8000 Aarhus C
Denmark
e-mail: *brian@daimi.au.dk*

Zbigniew Michalewicz*
Department of Computer Science
University of North Carolina
Charlotte, NC 28223
USA
e-mail: *zbyszek@uncc.edu*

Abstract

The paper investigates a new Patchwork model for structured population in evolutionary search, where population size may vary. This model allows control of both population diversity and selective pressure, and its operators are local in scope. Moreover, the Patchwork model gives a significant flexibility for introducing many additional concepts, like behavioral rules for individuals. First experiments allowed us to observe some interesting patterns which emerged during evolutionary process.

Category: artificial life, agents

1 Introduction

In this paper we investigate a new model, which combines properties of the island and diffusion models; moreover, a structured population of this model may vary in size. This Patchwork model allows great flexibility in introducing additional concepts, like self-organized criticality, age of individuals, migration, speciation, and it can be used for both stationary and non-stationary environments. The idea of the Patchwork approach is to introduce biological concepts of ecology and population biology to evolutionary algorithms. The modeling technique is closely related to multi-agent systems, where a system is represented by autonomously interacting entities. One major difference between Patchwork model and traditional evolutionary algorithms is the representation of individuals. In the Patchwork model, individuals are represented as autonomous mobile agents, which live in a virtual ecological niche and interact with their environment through their sensors and motors. Their decisions are based on local information that they collect with their sensors and their actions solely affect their local environment. Further, each individual has specific properties such as its maximum life span, ability to breed, (fitness dependent) mortality, and preferences in decision-making. Another important design feature of the Patchwork model is the spatial structure of the population, i.e. agents move and interact in a two-dimensional grid space. Each grid cell (called a patch) has local spatial properties, such as the maximum number of individuals it can carry.

Additional motivation for this work was based on the observation that the issue of self-adapting values of parameters of an evolutionary algorithm is one of the most important and promising areas of research in evolutionary computation, as it has a potential of adjusting the algorithm to the problem while solving the problem.

The paper is organized as follows. The following section discusses briefly the related work on population size and structured population. Section 3 presents some intuitions behind the new model, whereas section 4 defines the details of the model. Section 5 provides the implementational details of the simplified Patchwork model and gives the results of some experiments. Section 6 concludes the paper.

2 Related work

Several researchers have investigated the size of population for genetic algorithms from different perspectives. Grefenstette [15] applied a meta-GA to control parameters of another GA (including populations size and the selection method). Smith [26] proposed an algorithm which adjusts the population size with respect to the probability of selection error. Arabas et al. [1]

* also at Institute of Computer Science, Polish Academy of Sciences, ul. Ordona 21, 01-237 Warsaw, Poland

investigated a genetic algorithm with varying population size (GAVaPS); this algorithm introduced the concept of "age" of a chromosome, which is equivalent to the number of generations the chromosome stays "alive". In this approach the age of the chromosome replaces the concept of selection and, since it depends on the fitness of the individual, influences the size of the population.

As all evolutionary algorithms maintain a population of solutions, they are parallelizable in a natural way. As we mentioned in the Introduction, a population of individuals can be structured in different ways. The main two categories include the island model, where several sub-populations evolve in parallel and the diffusion model, where individuals are partitioned across the processors. Research on these models concentrated on the specific issues which are unique to these models. For the island model these included number and sizes of sub-populations, their communication topology, number of migrants and epoch lengths (fixed or variable), migrant selection strategies, etc. For the diffusion models [23] these included techniques for selecting parents, recombination techniques, size and shape of demes, etc. [23].

Note, however, that both the island model and diffusion model assume a fixed size of the population. In the diffusion model the population size is constant; most of the research was connected only with sizes of the neighborhood. In the island model, usually each of N sub-populations had a fixed number $\mu = M/N$ individuals. Of course, in general, each sub-population may have different size of μ_i (where $\sum_{i=1}^{N} \mu_i = M$). The migration process may decrease the size of one sub-population and increase the size of some other population (thus keeping M constant), but in most implementations μ_i's are kept constant (as a new migrant replaces one of existing individuals). However, the size of the population is one of the most important choices for the evolutionary search, since it influences population diversity and selective pressure. The proposed PATCHWORK model tackles this problem by providing adaptive population size; it combines the ideas of island and diffusion models, allowing also introduction of additional concepts, taken from the research on multi-agent environments and artificial life.

Since Darwins *The expression of emotion in man and animals* [8], people have been aware of the major role emotions play in behaviour. Particularly since Damasios *Descartes error* [9] people have been aware of the major role emotions play in decision-making. Naturally there has been much modeling of emotions in the multi-agent and individual based modeling communities, also much argumentation for the necessity of emotions for "believable" robots and software agents. Typical accessible articles with reference to further literature are [2, 7]. However, today's biologists consider motivation as the base of decision making [21]. Usually motivation is defined as a composition of shifting priorities for different behavioral drives.

The other major concept we borrow from the biological, multi-agent and individual based modeling communities is that of adaptation to degenerating environments. Our individuals adapt by moving to a new patch in the patchwork (hence the name of the proposed model). Biological individuals have also several other mechanisms: hypermutation, changing from asexual to sexual reproduction [13].

The idea to model the evolution of autonomous agents in an ecological niche has been introduced in ALife and ecology. One general purpose approach in ALife is called SWARM [22], which basically consists of Object C libraries for modelling of artificial ecology. In ecology, so called individual-based models have become very popular in the past years [18]. However, only very few approaches truly represent individuals as autonomous agents and mimic evolution [20, 10].

The final aspect present in the PATCHWORK model is that of self-adaptation. The issue of controlling values of various parameters of an evolutionary algorithm is one of the most important and promising areas of research in evolutionary computation: it has a potential of adjusting the algorithm to the problem while solving the problem [11]. The PATCHWORK model incorporates some "self-adaptation ideas" [16, 25] as an agent carries additional chromosomes (apart from a solution chromosome) which determine its behavioral patterns; the values present in these additional chromosomes are self-adaptive. The ideas behind the PATCHWORK model are discussed in the following section of the paper.

3 The idea

In all implementations of diffusion models for evolutionary algorithms there was a one-to-one mapping between individuals and processors: the number of individuals is equal to the number of processors (consequently, the population size remains constant at all times). If the interconnection topology of a diffusion model is a grid (Fig. 1(a)), a single individual corresponds to each grid cell. In the PATCHWORK model (for grid interconnection topology) a variable number of individuals corresponds to each grid location (Fig. 1(b)).

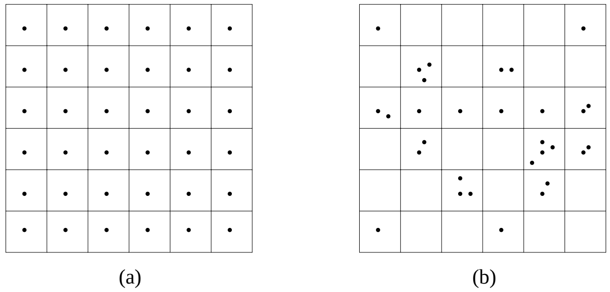

Figure 1: A grid interconnection topology in classical diffusion models (a) and in the PATCHWORK model (b)

This simple modification triggers many consequences. First of all, on each grid location we may have zero, one, or more (up to η) individuals; thus the size of the population is variable with maximum value of ηg^2 (for a grid $g \times g$). Second, the model can be considered as a special version of the island model, where each sub-population occupies one grid location: additional rules would define migration policies (from one grid location to another). These rules would imply fixed interconnection topology with adaptive flow of individuals between grid cells, which defines where and when an individual would move. Additionally, in contrast with the island model, some sub-populations can stay empty for some number of generations. Third, due to a clear structure of the whole population, the model borrows heavily from the diffusion model: the metric of the grid plays important part in various actions of the population. However, in contrast to other diffusion models, an individual can migrate from one location of a grid to some other location.

The main aspect of the PATCHWORK model is, however, that each individual is modeled as a mobile agent that act in a two-dimensional virtual world. These mobile agents are extended by a set of adaptive behavioral rules; they interact with each other only within grid cells. These rules determine the actions of individuals; e.g., their desire to reproduce or migrate. Each individual makes a bid for its desired action; the action may or may not take place. In other words, the concurrent events are handled by a two step process: first, agents make decisions and schedule actions; and second, the simulation shell executes the scheduled actions and resolves temporal conflicts. For instance, when agents within the same grid cell decided to mate, their intention is first scheduled, but not immediately executed. Here, a bid for reproduction might be unsuccessful because of lack of a partner or because there is no room for an offspring; even if offspring is produced, it may not be competitive enough to be included in the population. Similarly, a bid for migration might be unsuccessful due to bids of other individuals and/or over-crowding.

4 The model

Each individual (agent) in the population consists of two parts: its genome and so-called *motivation network*. Its genome is represented as a set of three chromosomes: (1) solution chromosome, (2) chromosome of standard deviations, and (3) chromosome of parameters. The solution vector $\vec{x}$ is used for calculating the fitness of the agent; we assume the existence of an objective function: $eval(x_1, \ldots, x_n) \rightarrow \boldsymbol{R}$ and the fitness F of an agent depends only on $eval$. The fitness function F takes values from the range $[0, 1]$; the higher the fitness, the better the agent. The second vector $\vec{\sigma}$ is a vector of standard deviations: mutations are realized by replacing $\vec{x}$ by $\vec{x}^{t+1} = \vec{x}^t + N(0, \vec{\sigma})$, where $N(0, \vec{\sigma})$ is a vector of independent random Gaussian numbers with a mean of zero and standard deviations $\vec{\sigma}$. The third vector $\vec{p}$ plays important part in the motivation network of an agent. A *motivation network* (which mimics the decision-making process of animals) [19] consists of (i) a set of mapping functions f_i, (ii) a set of operations op_i, (iii) motivation variables mv_j, (iv) a decision-maker, and (v) behavior patterns bp_k (Fig. 2).

There is a set $\vec{f}$ of predefined functions f_j $(1 \leq j \leq q)$; these functions (so-called *mapping functions*) are used for determining the behavior of the agent (more precisely, they define so-called *motivation variables* of the agent). Each of the function f_i takes up to s parameters (the values of these parameters are given in the vector $\vec{p} = (p_1, \ldots, p_s)$) plus an additional input $\mathcal{I}$ (the input $\mathcal{I}$ may include the values taken from the agent's sensors, internal state of the agent, etc). Thus $f_j(p_1, \ldots, p_{s_j}, \mathcal{I}) \rightarrow [0, 1]$, for $j = 1, \ldots, q$, where $s_j \leq s$. Motivation variable mv_i is defined by an operation op_i applied to a subset of functions f_j's.

In general, at each time step, the network receives information from the input variables, i.e., stimuli from sensors and internal states, which is mapped to the motivation variables (these input variables are included in $\mathcal{I}$). Note that there is a many-to-many relationship between mapping functions f_i and motivation variables mv_j. The value of each motivation variable ranges from 0.0 (no motivation) to 1.0 (maximum motivation) and is given by one or more mapping functions, which are specified by a subset of the agent's genes. Finally, the decision-maker determines and schedules a behavior pattern according to the motivation variable with the presently highest value. In other words, a motivation network selects an action from a behavioral repertoire (e.g., mating, moving, figthing, etc).

An agent interacts with its environment in the following way: Behavioral genes $\vec{p}$ together with input variables $\mathcal{I}$ determine the values of mapping functions f_i's, which, in turn, define the values of motivation variables mv_j. Then a decision maker selects a behavior pattern bp_k (see Fig. 2), which may or may not be executed (a decision is made during the conflict resolution stage on the basis of the agent's fitness).

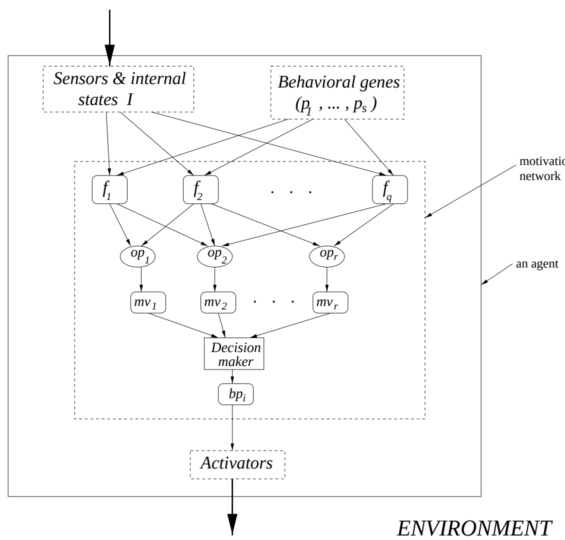

Figure 2: The structure of a motivation network. Motivation variables mv_i are defined by an operation op_i applied to a subset of mapping functions $f_1(\vec{p},\mathcal{I}),\ldots,f_q(\vec{p},\mathcal{I})$; a "decision maker" selects one of these variables as a "behavior pattern"

The outline of the control structure of the PATCHWORK model is given in Fig. 3. The PATCHWORK model maintains a population of agents, $P(t) = \{a_1,\ldots,a_{n(t)}\}$ for iteration t. Each agent represents a potential solution to the problem (in terms of its solution vector $\vec{x}$, which determines also some measure of its "fitness"). Each agent has a specific location (x, y) on the grid. A new population (iteration $t+1$) is formed by following a sequence of steps. First, a behavior pattern for each agent is determined by its motivation network. Second, for each grid location, each behavior pattern is considered in turn (the order of behavior patterns is fix and remains the same for the whole process). The conflict-resolution procedure selects more fit agents for performing their desired action. For example, if two agents desire to move to a particular location (which can accommodate only one new agent), the fitter one is selected. Then, for each

```
procedure PATCHWORK Model
begin
    t ← 0
    initialize P(t)
    evaluate P(t)
    while (not termination-condition) do
    begin
        t ← t + 1
        for all agents in the population do
            determine agent's behavior pattern
        for each grid location (x, y) do
            for each behavior's pattern bp_i do
                resolve_conflicts
                for each agent at (x, y) with bp_i do
                    perform_action
        remove agents who died from P(t)
        evaluate P(t)
    end
end
```

Figure 3: The structure of the PATCHWORK model evolutionary algorithm

agent within the grid location with the same behavior pattern, an appropriate action is performed. For example, the behavior pattern of "mating" would trigger a specific sequence of actions (selection of a partner, application of variation operators—crossover and mutation, placement of an offspring in this grid location, if possible), whereas the pattern of "moving" triggers some other sequence of actions (determination of direction of the move and move itself, if possible). Note that in the PATCHWORK model, the central component of any evolutionary algorithm, the selection process, is hidden here in the "resolve_conflicts" and "perform_action" components, as all decisions whether an agent makes a successful action (e.g., mating or moving) are resolved there.

Note that the order of behaviors' patterns in the loop "**for** each behavior's pattern bp_i **do**" is important. Due to the conflict resolution module, the development of the population depends on whether the behavior "moving" is considered before or after "mating", as new immigrants (or new offspring) can take last available slots in a grid location.

The reproductive success of an agent is another important aspect of the model. It is crucially affected by its probability of survival (i) as an adult and (ii) as a juvenile. Adult agents die either when they reach a maximal age (*age*) or earlier, with a probability in respect to their specific mortality m (death rate), which is negatively correlated with their fitness. This design corresponds to natural selection in real biology. However, the precise design of the correlation model is crucial. In fact, it could happen that the population either might crash and get extinct (too high mortality) or would quickly become very crowded (too low mortality). So, how do we determine an appropriate mapping from fitness to mortality? In nature, populations expand until they reach the carrying capacity of their environment, i.e., the maximum number of individuals that can survive in a certain area due to limitation of resources or increase of predators or parasites. For instance, the maximal population density of predators is food constrained by the number of available prey items. Thus, natural systems are able to auto-regulate mortality by density dependent factors. However, this mechanism works only when ecosystems are close to an equilibrium. Rapid changes of the environment can significantly reduce or enhance the mean fitness of a species and might either lead to extinction or population explosion. One solution for this problem can be found in analytical models of constrained population growth. For example, mortality might be population density dependent: $m = pop\_size/\eta g^2$, i.e., the ratio of the current population size and the maximal population size.

On the other hand, the mortality of juveniles is determined as follows. During mating, agents are grouped as parents according to their fitness. After recombination of the parental genes, offspring is introduced to the local grid cell. Each couple is allowed to produce a fixed number of offspring. Afterwards, we rank all new local juveniles according to their fitness and kept as many of the best as there is space left.

5 Experiments with a simplified model

In this section we describe (1) some implementational details of a simplified PATCHWORK model, and (2) the results of some experiments performed on two landscapes.

In all experiments reported in this paper the following assumptions were made: (1) the size of the grid structure is 9×9, i.e., $g = 9$, (2) the maximum number of agents per grid location is 14, i.e., $\eta = 14$. Thus the maximum number of agents in the whole environment is $14 \times 81 = 1134$, however, the population was initialized with 200 agents, (3) the number of variables of solution vector $\vec{x}$ is 2, i.e., $n = 2$, (4) the vector of standard deviations $\vec{\sigma}$ is removed; Gausian mutation was applied with an initial standard deviation of 4, which linearly declined to 0.001 after 600 simulation time steps (as opposed to self-adaptive value), (5) the parameter vector $\vec{p}$ is either empty (first experiment) or it consists of a single parameter (second experiment); in the latter case it takes a random value from $[0, 5]$, (6) the set of input variables $\mathcal{I}$ for an agent a consists of one variable y (i.e., $\mathcal{I} = \{y\}$); its value is equal to the number of other agents present in the same grid location as the agent a. Thus $0 \leq y \leq \eta - 1 = 13$, (7) the maximum age of an agent is 4 generations, i.e., $age = 4$, (8) there are two motivation variables only: *mating* and *moving*, (9) the order of behavior patterns is fixed: the behavior "mating" is considered before "moving", (10) the mortality rate m is the average of two measures: $m = (m_1 + m_2)/2$, where m_1 is determined by the fitness (i.e., the value of the objective function) of the agent: $m_1 = \frac{M - F(\vec{x})}{M}$, where M is the maximum objective value and m_2 is determined by population density at time t: $m_2 = \frac{pop\_size(t)}{\eta \cdot g^2}$, (11) mating and reproduction occurs only within the local environments, i.e., the grid cells. An agent that decided to mate has reproductive success if (i) it can find a mate, (ii) there is space for more individuals in its grid cell, and (iii) its offspring has a higher fitness compared to offspring of other agents from the same space. Mating agents recombined their genetic code by standard operators: 1-point crossover and Gaussian mutation, (12) the decision to move results in a movement to a neighbored cell. Moving directions are initially random, but are gradually changed by a small random angle (in the current version, from the range $(-\pi/4, \pi/4)$) before an agent moves, (13) the movements are not performed when agents try to cross the boundary of the virtual world (however, it would be straightforward to modify this by assuming toroidal grid structure).

In this simplified version of the PATCHWORK model, the motivation network was fixed for all agents. There are two predefined functions (see Fig. 2):

$$f_1(y) = \max\{0, \min\{1, y\}\}, \text{ and}$$
$$f_2(p_1, y) = \max\{0, \min\{1, 1 - p_1 \cdot y/\eta\}\}.$$

The values of motivation variables were determined as

$$mv_{mate} = \min\{f_1(y), f_2(p_1, y)\}, \text{ and}$$
$$mv_{move} = 1 - \min\{f_1(y), f_2(p_1, y)\}.$$

Thus, the motivation of an agent to mate is 1, when one other agent is in the same local patch, and drops linearly with the number of local neighbours. When no local agents are present, the motivation to mate is zero.

In each patch, the number of potential offspring was equal to the number of local agents. Agents with a higher fitness had proportionally more matings. The best offspring were inserted in available positions (if any) in the grid. All runs were made for 1000 time steps. The results presented in section 5.1 are mean values of 50 repeated runs per experiment.

5.1 Results

We have tested the performance of a simplified PATCHWORK model on two landscapes. In each of these scenarios, agent's behaviour was either (i) prespecified or (ii) included a self-adaptive component.

It is interesting to investigate the relationship between values of motivation variables during a run: at different stages of evolutionary process, an agent may have better motivation to mate or to move. Figure 4 shows a characteristic shift of an agent's motivational state from mating to moving at time $t = 26$ due to a change from a small number to a large number of other agents at the grid location of the agent.

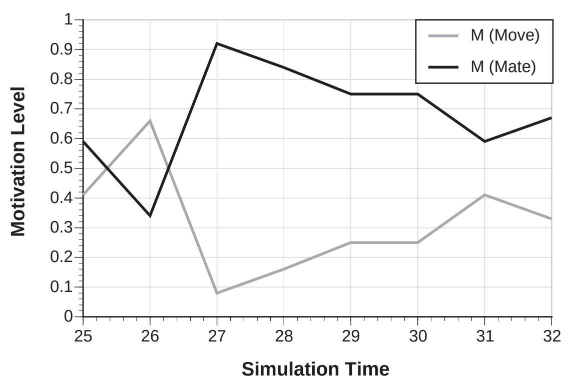

Figure 4: The values of motivation variables for moving and mating at time $t \in [25, 32]$

Figure 5 shows our first test function:

$$eval_1(x_1, x_2) = (50 - |x_1 - 50| + 40 sin(\tfrac{5}{18}\pi x_1)) + (50 - |x_2 - 50| + 40 sin(\tfrac{5}{18}\pi x_2)),$$

where $0 \leq x_i \leq 100$ for $i = 1, 2$. Its global maximum value of 175.63284 is at $(x_1, x_2) = (52.167, 52.167)$.

The PATCHWORK model with prespecified agent behaviour reached the fitness optimum 175.63284 after 614 time steps (Fig. 6). The population size was stable (around 500 agents, i.e., less than 50% of the allowed maximum population size (Fig. 7)).

The model with self-adaptive behaviour was of comparable quality, but operated with a higher mean population size. During the run, the gene p_1 controlling

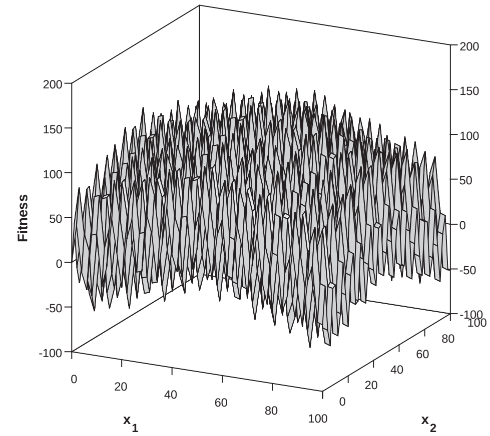

Figure 5: The landscape of objective function $eval_1$

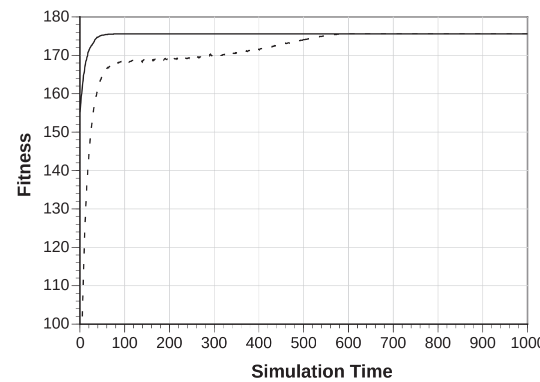

Figure 6: Prespecified agent. The objective value of the best individual (continuous line) and the population mean (broken line) over simulation time

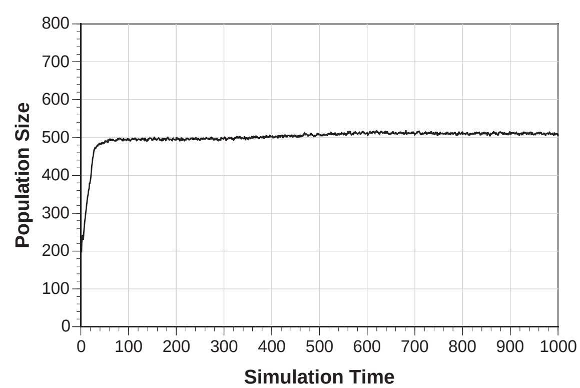

Figure 7: Prespecified agent. Population size versus simulation time

the slope of the motivation level function for mating decreased from a mean of 2.50 at $t = 0$, to 0.73 at $t = 100$, to 0.59 at $t = 1000$. This change influenced the distribution of values of motivation variables for different number of neighbors. Hence, agents tried to mate more frequently towards the end of the runs. This effect is also visible in Fig. 8, which shows a series of close-up views to the arena of the patchwork at different simulation times for agents with evolving decision-making. In the beginning, agents explored the arena (filled circles). After a while, most patches are occupied by agents. Finally, agents mainly preferred to mate (small rectangles) instead of moving (circles).

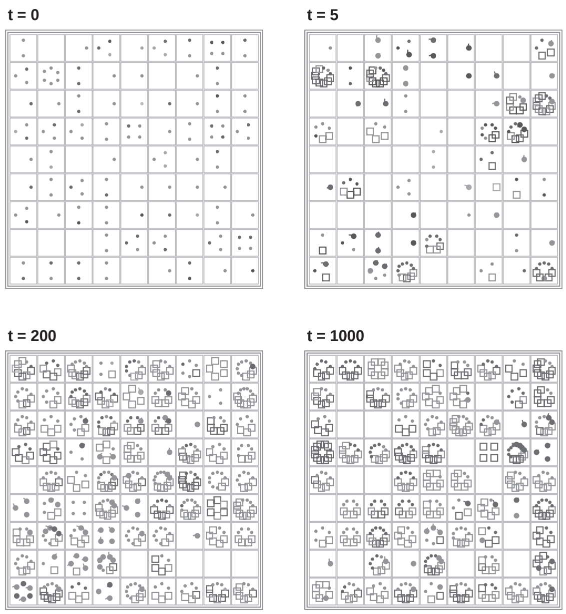

Figure 8: The arena at different time steps. Small circles indicate offspring, large circles – moving adults, squares – mating adults. Agent brightness corresponds to fitness, i.e., the darker the color the higher the fitness

Another test-function was the Schaffer's function $F6$:

$$eval_2(x_1, x_2) = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1.0 + 0.001(x_1^2 + x_2^2)]^2},$$

where $-100 \leq x_i \leq 100$ (Fig. 9). This function has a global minimum value of 0 at $(x_1, x_2) = (0, 0)$.

Here, optimisation process took longer, but got close to the optimum at $t = 120$ for the fixed behaviour case (Fig. 10) and at $t = 200$ for the self-adaptive behaviour case. In both cases, population sizes stayed stable (Fig. 11). As in the previous case, agents preferred to mate more frequently as simulation time increased, though not as much as for test-function $eval_1$.

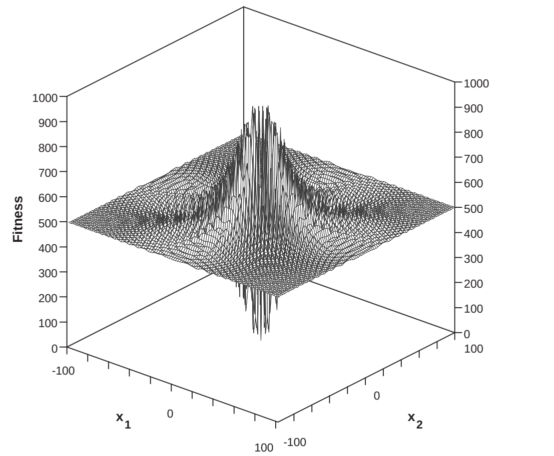

Figure 9: The landscape of objective function $eval_2$

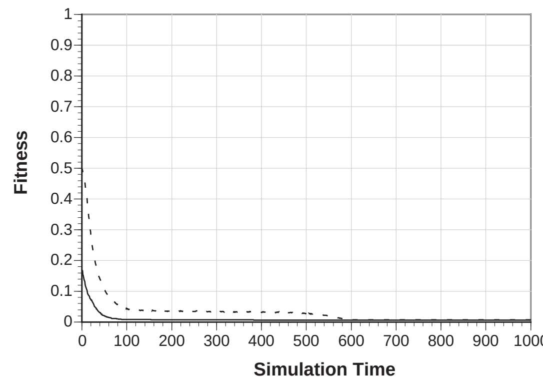

Figure 10: Prespecified agents. The fitness value of the best individual (continuous line) and the population mean (broken line) over simulation time

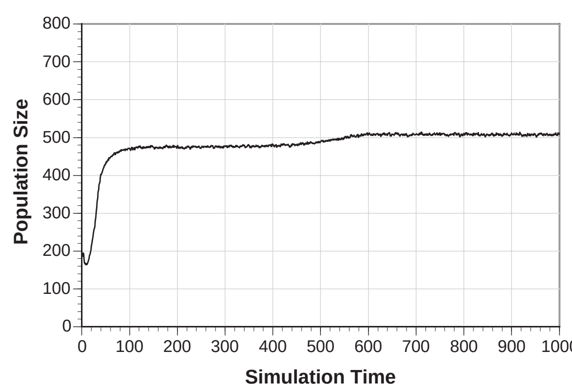

Figure 11: Prespecified agents. Population size versus simulation time

6 Conclusions

The experiments demonstrated that the system converges to the optimum (on selected two landscapes); during the run it "controls" its population size. It would be interesting to investigate further significance of various parameters of the model, e.g., the influence of mortality rate m on the results, etc. Also, the next version of the model would include motivation network with self-adaptive components at all levels; this version should be also useful for non-stationary environments.

Further, the proposed PATCHWORK model generalizes previously proposed models for parallel implementations of EAs. Additionally, an agent in this model consists of two parts (genome and motivation network), which allow (1) exploration of self-adaptive capabilities of an agent and (2) investigation on behavior patterns of the agents. As a side effect, the population size varies during the run. The model could be used to investigate the effect of additional sensors and behavior patterns. For instance, "moving" is beneficial when (i) no mates are available or (ii) the available space for breeding is very limited. Interestingly, the encounter of individuals with a high fitness had different effects. An individual with a high fitness should stay, because it is competitive enough to mate with another individual of high fitness. On the other hand, an individual with a low fitness should stay only if the local population density is very low. An appropriate behavior would require a sensor which could perceive the fitness of other individuals.

References

[1] Arabas, J., Michalewicz, Z., and Mulawka, J., *GAVaPS — a Genetic Algorithm with Varying Population Size*, Proc. of the 1st IEEE ICEC, 1994, 73–78.

[2] Aube, M., Senteni, A., *Emotions as Commitment Operators: A Foundation for Control Structure in Multi-Agent Systems*, Springer, LN AI 1083, 13–25, 1995.

[3] Bak, P., *How Nature Works*, Oxford Univ. Press, London, 1996.

[4] Bäck, T., Hoffmeister, F., and Schwefel, H.-P., *A Survey of Evolution Strategies*, in [6], 2–9.

[5] Bäck, T., Fogel, D.B., and Michalewicz, Z., *Handbook of Evolutionary Computation*, Oxford Univ. Press, New York, 1996.

[6] Belew, R. and Booker, L. (Editors), Proc. of the 4th ICGA, Morgan Kaufmann, San Mateo, CA, 1991.

[7] Boteilho, L. M., Coelko, H., *Emotion-Based Attention Shift in Autonomous Agents*, Springer, LN AI 1193, 277-292, 1996.

[8] Darwin, C., *The Expression of Emotion in Man and Animals*, London J. Murray, Chicago, Univ. Chicago Press, 1965.

[9] Damasio, A. R., *Descartes Error: Emotion, Reason and Human Brain*, Picador, London, 1994.

[10] Downing, K. *EUZONE: Simulating the Evolution of Aquatic Ecosystems*, Artificial Life, 3, 307–333, 1997.

[11] Eiben, A.E, Hinterding, R., and Michalewicz, Z., *Parameter Control in Evolutionary Algorithms*, submitted for publication, 1999.

[12] Forrest, S. (Editor), Proc. of the 5th ICGA, Morgan Kaufmann, San Mateo, CA, 1993.

[13] Funch, P. *Cycliophora is a new phylum with affinities to Entoprocta and Ectoprocta*, Nature, 368, 711–714, 1995.

[14] Gorges-Schleuter, M., *ASPARAGOS An Asynchronous Parallel Genetic Optimization Strategy*, in [24], 422–427.

[15] Grefenstette, J.J., *Optimization of Control Parameters for Genetic Algorithms*, IEEE Trans. on Systems, Man, and Cybernetics, 16(1), 122–128, 1986.

[16] Hinterding, R. and Michalewicz, Z. and Eiben, A.E., *Adaptation in Evolutionary Computation: A Survey*, Proc. of the 4th IEEE ICEC, 65–69, 1997.

[17] Hinterding, R., Michalewicz, Z., and Peachey, T.C., *Self-Adaptive Genetic Algorithm for Numeric Functions*, Proc. of the 4th PPSN, 1996, Springer, LN CS 1141, 420–429.

[18] Judson, O. P. *The rise of the individual-based model in ecology*, TREE, 9, 9–14, 1994.

[19] Krink, T. and Mayoh, B., *Motivation Networks - A Biological Model for Autonomous Agent Control*, submitted to IJCAI-99, Stockholm, submitted.

[20] Krink, T. and Vollrath, F. *Analysing spider web-building behaviour with rule-based simulations and genetic algorithms*, Journal of Theoretical Biology, 185, 321–331, 1997.

[21] Manning, A. and Dawkins, M., *An introduction to animal behaviour*, Cambridge: Cambridge University Press, 1998.

[22] Minar, N., Burkhart, R., Langton, C., and Askenazi, M. *The Swarm Simulation System: A Toolkit for Building Multi-Agent Simulations*, Working Paper, The Santa Fe Institute, rep. no.: 96-06-042, 1996.

[23] Pettey, C.C., *Diffusion (cellular) models*, in [5], section C6.4.

[24] Schaffer, J., (Editor), Proc. of the 3rd ICGA, Morgan Kaufmann Publishers, San Mateo, CA, 1989.

[25] Smith, J.E., *Self Adaptation in Evolutionary Algorithms*, Ph.D. Thesis, University of the West of England, Bristol, 1997.

[26] Smith, R.E., *Adaptively Resizing Populations: An Algorithm and Analysis*, in [12], 653.