Journal of Computational Science 10 (2015) 351–359

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

# Evolving agent-based models using self-adaptive complexification

Michael Wagner[a,*], Wentong Cai[b], Michael H. Lees[c,d], Heiko Aydt[a]

[a] TUM CREATE, 1 Create Way, Singapore 138602, Singapore
[b] School of Computer Engineering, Nanyang Technological University, Singapore 639798, Singapore
[c] Computational Science, University of Amsterdam, Netherlands
[d] ITMO University, St. Petersburg, Russian Federation

ARTICLE INFO

*Article history:*
Received 13 October 2014
Received in revised form 27 February 2015
Accepted 20 March 2015
Available online 24 March 2015

*Keywords:*
Evolutionary computing
Agent-based simulation
Modeling
Artificial life

ABSTRACT

This paper focuses on parameter search for multi-agent based models using evolutionary algorithms. Large numbers and variable dimensions of parameters require an optimization method which can efficiently handle a high dimensional search space. We are proposing the use of complexification as it emulates the natural way of evolution by starting with a small constrained search space and expanding it as the evolution progresses. To further improve performance we suggest and experiment with methods of self-adaptation to enable the algorithm to adjust its parameters individually to the problem at hand. We examined the effects of these methods on an EA by evolving parameters for two multi-agent based models.



## 1. Introduction

Agent-based models (ABMs) are among the most important tools for exploring emergent behavior, a phenomenon that describes the behavior of a system, which cannot be explained alone by the sum of its parts. Understanding and harnessing emergence is very important because it allows to create complex behavior, based on the interaction between relatively simple components. One way to discover and examine emergence in a computer simulation is to calibrate the model parameters accordingly. Underlying models often encompass a wealth of parameters, making the search of the sheer size of multi-dimensional space a problem. Due to interdependence and interaction between agents, slight changes in the model configuration can amount to very different simulation outcomes, indicating the high level of complexity. Even though evolutionary algorithms (EAs) are often designed and used to efficiently explore large parameter spaces, traversing those can still take a considerable amount of time. In this paper we propose the use of complexification to improve the performance of EAs used for parameter estimation of multi-agent based models. In an earlier work [12] we gave evidence that evolving parameters is directly influenced by the model's complexity. Therefore it is essential for EAs to be flexible enough to handle complex models. Traditional EAs are forced to make assumptions about the problem. Properties like the length and structure of genomes have to be determined a priori and cannot be changed during the EA run. However, complex systems can have a variable number and structure of parameters. Rule-based ABMs face the same issues. Rules can consist of an arbitrary number of components like conditions and actions. This makes finding optimized solutions very difficult if those require a large number of components or complex rule sets. Natural evolution is far more than just a cycle of recombining and mutating genes. In order to advance from basic single-cell lifeforms to complex organisms, the genotype of species has to be extended [3,4]. Complexification is a substantial part of this process as it allows for an organism to increase its genome size and to become more versatile. By incrementally adding new genes organisms can, over time, adapt to their environment and develop further characteristics or skills. In nature this happens by gain-of-function mutation, a type of mutation that increases the genome size and confers new properties and eventually new functionality to the organism. The increase in size most commonly happens by a random duplication of parts of the genome. By using complexification repeatedly the evolutionary process can cover a wider variation of potential traits and functions of the individuals subjected to it. We believe that complexification, as a form of incremental evolution, is able to improve the estimation of model parameters. We want to validate this by conducting experiments on two multi-agent based models where we test an evolution strategy against a redesigned version of itself which incorporates complexification. This allows for a direct comparison of the evolutionary techniques. Extending the EA to make use of complexification comes at the cost of introducing more parameters. To mitigate this we propose and

* Corresponding author. Tel.: +65 98393967.

http://dx.doi.org/10.1016/j.jocs.2015.03.005
1877-7503/

experiment on mechanisms for self-adaptation of the additional parameter by the EA itself. In the following parts of our work we first provide a short overview on the evolution of model parameters for ABM and complexification. Secondly, in Section 3 the benchmark models for testing our complexifying EA are introduced, followed by a description of the EA and its mechanics itself in Section 4. Finally in Section 5 our experiments are described and concluded with a discussion.

## 2. Related work

Evolution of model parameters in multi-agent based models has been pioneered in [6,13], among others. It is very useful as a tool to configure and explore the real life systems, which the models are based on.

One of its major applications of it is to detect and explore emergent behavior that may arise. Emergent behavior is a phenomenon that requires and indicates the level of complexity of a natural or artificial system. In connection with ABMs it has recently been experimented with by [11]. In [12] the evolution of boid model parameters for discovering forms of emergence, in relationship to an objective fitness measure, was described. Furthermore it was discussed how the model complexity influences this process and what challenges it holds. We learned from this work that, while leaving the fitness measure unchanged, extending the model detail and thus increasing the number of possible configurations may have negative effects on the parameter evolution.

Our work continues this train of thought and argues how the apparent difficulties can be approached. Complexification with regard to EAs has been applied almost exclusively in evolving artificial neural networks (ANNs) [9], benefiting from their easily modifiable graph-like structure. Other applications include the evolution of strategies for single 3D agents [10] successfully proved the superiority of incremental evolution by evolving the weights rather than structure of the ANN that is controlling the agents movement.

## 3. Agent-based models for experimentation

We chose two multi-agent based models to test our hypothesis: the boid model, introduced in [5] and its further development, the bee swarm model. The models we use are able to represent multiple species of agents to study cooperative and competitive behavior, which are important factors in the creation of emergence. The emphasis is put especially on their effect on the survival rate of the boids.

### 3.1. Boid model

Our customized boid model, as described in [12], contains a simple predator-prey scenario. Additionally to the boids, which are now considered prey, it also involves predators and food sources for the boids, thus creating a simple food chain. The goal for which to optimize the model parameters, is to maximize the boid survival chance by having them graze the food sources and evade the predators as efficiently as possible. To add more strategic depth, the boids can be grouped in up to three different species where all members of one species have identical parameters. However, different species can have different parameters. This is a generalization of cooperation between multiple species as it occurs in nature among certain kinds of birds [7], among others. These species ignore each other by default when it comes to flocking behavior, but there are flags, called "alliances", which can be set to enable collective flocking. Boid species have two options: either cooperate or to try to survive on their own. For this model we define a fitness function $f_{survived}(\vec{x})$ in Eq. (1) as the number of boids which are still alive after the simulation terminates at simulation time $t_{stop}$. The argument $\vec{x}$ hereby denotes the parameter vector that will be used to initialize the simulation.

$$f_{survived}(\vec{x}) = |b \in aliveBoids_{t_{stop}}| \tag{1}$$

The challenge now is to find the following parameters, contained in vector $\vec{x}$, so that $f(\vec{x})$ is optimized:

1. The number of different boid species (in this case ranging from 1 to 3).
2. The five parameters regarding movement and flocking behavior: avoidance, convergence, cohesion, momentum and sensor range.
3. The alliance flags between the boid species.

### 3.2. Bee swarm model

We transformed the boids to a swarming model and added dependencies between them to make it more suitable for exploring complexification. According to its name, the model attempts to give a simplified presentation of how a bee swarm works. Firstly, the bees all originate from a hive, centered in the simulated world. In order to sustain their colony it is necessary to look for food, forage it and return it to the hive. The bees are facing two challenges here: food sources e.g., flowers, vary in the amount of food they provide and there are competitors present, eager to steal food from the hive. Similar to the boid model, bees can be also divided into species. This emulates the natural division of labor into workers and drones. In contrast to nature, the groups can choose their specialization by distributing points into three different skills: foraging, scouting and defending. Each skill may have between 0 and 10 points invested and has been artificially equipped with artificial caveats and benefits to create trade-offs and avoid the generation of perfect boids. Foraging skill increases carrying capacity but penalizes defense while the defending ability acts conversely. Scouting ability benefits sensory range and movement speed, but takes away from both, defense and foraging skills. These are the most important evolvable parameters in our model as they strongly influence the bees behavior. The number of bee groups as well as their share in the total bee population are two more evolvable parameters. Remaining parameters are the following five, similar to the predator-prey boid model: avoidance, convergence, cohesion, momentum and size of neighborhood. The challenge here is to find the following parameters, so that a given fitness function $f_{gathered}(\vec{x})$ in Eq. (2) is optimized:

1. The number of different bee groups (in this case ranging from 1 to 3).
2. The share of bees per group.
3. The skill distribution per group.
4. The five parameters regarding grouping behavior, as mentioned above.

In our experiments we use a relative food gathering fitness $f_{gathered}(\vec{x})$, which expresses the relationship between gathered food and lost or spent food. Similarly to the boid model, $f(\vec{x})$ is to be maximized.

$$f_{gathered}(\vec{x}) = \frac{gatheredFood(hive)}{spentAndLostFood(hive)} \tag{2}$$

Fitness function of bee swarm scenario, relation between gathered and spent food.