Evolving Swarms that Build 3D Structures

Sebastian von Mammen
Lehrstuhl für Programmiersprachen
Institut für Informatik
Friedrich-Alexander-Universität
Erlangen-Nürnberg
Martenstrasse 3, 91058 Erlangen
sebastianvonmammen@web.de

Christian Jacob
Evolutionary & Swarm Design Group
Dept. of Computer Science
University of Calgary
2500 University Drive NW
Calgary, AB, Canada T2N1N4
jacob@cpsc.ucalgary.ca

Gabriella Kókai
Lehrstuhl für Programmiersprachen
Institut für Informatik
Friedrich-Alexander-Universität
Erlangen-Nürnberg
Martenstrasse 3, 91058 Erlangen
kokai@informatik.uni-erlangen.de

Abstract- The complex interactions of natural swarms, for example formed by some social insects, are difficult to comprehend. Considering tasks such as nest-building, the necessary underlying communication presumably happens indirectly by changing and reacting on the environment. This paper presents an overal approach to interactively evolve rule-based swarms that create three-dimensional structures in continuous space. The approach comprises the design of the swarm agent, details about the breeding process and first results. A swarm is determined by a set of flocking parameters and a set of instructional rules that allow the agents to change their local structural environment. The center or focus of the swarm's endeavour may be shifted either on a swarm agent or on a fixed point in space. The alteration of the supplied 3D structure during the course of evolution enables an external supervisor to interactively guide the development of a swarm.

1 Introduction

Craig Reynolds' flocks provide us with the possibilities to simulate an appropriately coordinated swarm movement [rey87]. Based on this model Kwong and Jacob have discovered several flight behaviours that are induced by different weightings of a swarm agent's steering urges [jac03]. Besides the coordination of movement social insect swarms also accomplish complex tasks such as nest-building [bon99]. There is only little knowledge about how insect swarms create 3D structures. Before new insights can be utilized, for example in the domain of self-assembly processes, swarm models and behaviours have to be designed and their capabilites have to be investigated.

Contrary to anthropomorphic approaches of building nests - which basically means that each agent follows a blueprint of an architecture - an insect's behaviour depends on local information and is determined by a probabilistic stimulus-response scheme. Communication between the swarm agents happens as an important by-product of this behaviour. One agent alters the environment and another one reacts to these changes. It is assumed that qualitative stigmergy, where an agent reacts on a discrete occurance of some sort, plays the dominant role in building complex nests. Discrete stimuli can be the construction elements that have already been built or the environmental structures that are provided by nature. Any occurance that initiates and guides a certain building behaviour is called a *template*.

Bonabeau et al. have designed rule-based lattice swarms which approximate the nest-building of eusocial wasps [bon99]. The agent's behaviour is determined by a set of rules whose preconditions consider the agent's local environment. Pilat reproduced these results and found additional rule sets that lead to other forms of wasp nest constructions and entirely new architectures [pil04]. His discoveries were accomplished by interactive evolution of swarms that built interesting structures.

This paper presents an approach of breeding similar rule-based swarms that act in a continuous world. The evolutionary process happens automatically by comparison between the swarms' construction and a pre-defined 3D structure. Another important new aspect is the parallel development of the flocking behaviour of a swarm along with its set of instructional rules.

The combination of an evolutionary algorithm with interactive evaluation is an appropriate choice to find new swarm behaviours that result in the construction of 3D structures. Interactive evaluation often leads to success, if there is only a vague notion of the concept's objective, the creative potential of the system is not yet fully investigated, or the emergent mechanisms of the resulting system are not easily reducible ([daw87], [sim91], [whi01], [tho02], [kwo03] and [jac01]).

This paper is organised as follows. The next section presents the swarm agents' functionality (genotype), comprising their flocking and construction behaviour, along with the simulation runs that compute a swarm's construction (phenotype). Section 3 describes the genetic operators that work on the swarm's representation and how a creative swarm emerges. Examples bred by swarms and a case study are shown in Section 4. Section 5 gives some ideas on extensions of further approaches.

2 The Swarm Simulation

One swarm consists of a number of equally acting agents. In order to find a swarm which produces interesting 3D structures, we evolve a population of swarms. First, a random flocking and construction behaviour is assigned to each member of this population. Then an appropriate swarm is bred by repeatedly computing the swarms' fitnesses and the consequent generation of a new population of swarms.

In our simulations the swarm agents are represented as small spheres[1]. A swarmette (= swarm agent) decides on a specific action whenever it collides with a cubic construction element. The fitness of a swarm is computed after the

[1] We used the VIGO swarm simulation library [bur04].

simulation has run for 500 virtual seconds. The proportions of the simulated world and the occuring objects are listed in Table 1.

Number of swarm agents	25
Spatial dimensions	$10 \times 10 \times 10$
Minimum distance between two construction elements	0.001
Construction element edge size	0.15
Swarmette sphere radius	0.05

Table 1: Proportions of the simulation

To initiate the building process, at least one construction element has to be provided. Figure 1 draws a scheme of the constructional process initiated by the collision between an agent and a construction element. An inherited behavioural rule checks the local environment for some structural characteristics. If they apply, which means that there are construction elements at certain positions relative to the collision location, the consequent action of a rule is exectued.

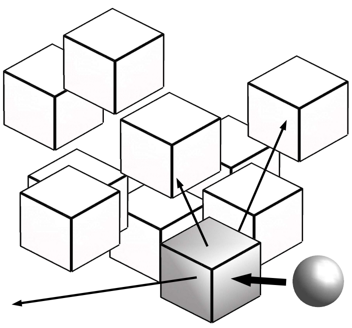

Figure 1: An agent, represented by a sphere, is going to collide with the grey cubic construction element (the agent's velocity vector is indicated by the thick arrow). The agent's behaviour depends on the structural configuration that surrounds the collision partner. In the illustrated case thin vectors point to the positions that will be checked by the agent for whether construction elements are present.

The set of possible actions after an agent-block collision is listed in Table 2. A construction element can be built at a location relative to the spatial coordinates of the construction element the agent has collided with. This relation is stated by a direction vector $\vec{d}$. There is a set D of default direction vectors according to the basic points of the compass (in detail these are $\vec{d}_{North}$, $\vec{d}_{South}$, $\vec{d}_{East}$, $\vec{d}_{West}$, $\vec{d}_{above}$, $\vec{d}_{below}$ and $\vec{d}_{here}$). The absolute values of these standard direction vectors are normalised to the construction element size, so that direction and distance of $\vec{d}_{above}$ point from one construction element exactly to its upper neighbour's center.

There are two types of destruction methods. The agent may destroy the construction element it has collided with. Another method is to destroy a construction element at a relative distance. The location of the construction element is computed by addition of a vector $\vec{d}_{destroy}$ to the collision construction element's coordinates $\vec{p}_c$. Of course, if there is no construction element at $\vec{p}_c + \vec{d}_{destroy}$, destruction does not take place.

The last class of actions refers to a center of the swarm which is not implicitly given but has to be explicitely announced. An agent can declare a specific location (including its own) as center of the swarm. If the urge towards the center is sufficiently high, a tendency towards the selected goal will appear. In this way all the agents have a common leader, if an agent has declared itself as center of the swarm. Alternatively, the swarm can concentrate its constructional measures on a specific region. If no center is defined, the flight of a swarm is not influenced by the center urge. In addition to its declaration an agent may recant the current swarm center.

Action	*Parameter*
Create a new construction element	Vector $\vec{d}$
Destroy the collision construction element	None
Destroy a remote construction element	Vector $\vec{d}$
Declare itself as the swarm center	None
Set the swarm center to a specific location	Vector $\vec{d}$
Recant the swarm center	None

Table 2: The rule-based agent's actuators. $\vec{d}$ is a location vector that starts in the center of the collision construction element and points to an arbitrary direction.

Aside from these collision dependent actions, the swarm agents continually alter their velocity according to several differently weighted steering urges. According to Reynolds' "boids" flocking model the agents are equipped with visual senses to perceive their neighbours [rey87]. Assume $\vec{d}_{si}$ as the vector between a swarmette s and another agent i of the swarm. Every agent i is in the neighbourhood N_s of a swarmette s, if the absolute value of $\vec{d}_{si}$ is within the swarmette's radius of perception r, and the angle α_{si} between the direction of s and the location of i is within some range $[0, 2]$ degrees radians. Which means:

$$\forall i \in N_s | (\|\vec{d}_{si}\| \leq r) \wedge (\alpha_{si} < 2). \tag{1}$$

At each time step of the simulation, the swarmette's velocity $\vec{V}_{vel}$ is updated with an acceleration vector $\vec{V}_{acc}$:

$$\vec{V}_{acc} = \sum_{j=0}^{5} w_j \vec{V}_j \tag{2}$$

w_j, $j \in [0..5]$ are the weights of the distinct tendencies during the flight. The absolute value of a swarmette's velocity can not exceed the maximum velocity V_{max} (set to 0.5) and the acceleration within one time step is limited to $A_{max} = 0.3$ (the given values were chosen to fit the simulation's dimensionality). The six urges that result in the swarm agent's acceleration are:

Center, $\vec{V}_0$: Considers the vector towards a fixed location within the simulation environment or to an agent's location. If no center is defined, $\vec{V}_0$ is zero.

Separation, $\vec{V}_1$: The direction away from all neighbours. As with the next two urges, a neighbour i's influence on agent s' flight decreases with growing distance $\|\vec{d}_{si}\|$. The division by the number of neighbours $|N_s|$ normalises the resulting vector:

$$\vec{V}_1 = -\frac{1}{|N_s|}\sum_{i \in N_s}\frac{\vec{d}_{si}}{\|\vec{d}_{si}\|^2} \tag{3}$$

Alignment, $\vec{V}_2$: Adjusting the agent's velocity to the average of the neighbours' velocities. The tendency is computed by subtraction of the swarm agent s' original velocity $\vec{v}_s$. Referring to each neighbour's velocity $\vec{v}_i$:

$$\vec{V}_2 = -\vec{v}_s + \frac{1}{|N_s|}\sum_{i \in N_s}\frac{\vec{v}_i}{\|\vec{d}_{si}\|^2} \tag{4}$$

Cohesion, $\vec{V}_3$: The center of gravity of the agent's neighbours. The tendency is computed by subtraction of the swarm agent s' original location $\vec{l}_s$. Referring to each neighbour's location $\vec{l}_i$:

$$\vec{V}_3 = -\vec{l}_s + \frac{1}{|N_s|}\sum_{i \in N_s}\frac{\vec{l}_i}{\|\vec{d}_{si}\|^2} \tag{5}$$

Ground, $\vec{V}_4$: Is responsible for the general tendency towards the ground. $height_s$ defines the height of the swarmette s and $height_{max}$ is the maximum height of the simulation environment:

$$\vec{V}_4 = -\left(0, \frac{1}{2}height_s/height_{max}, 0\right)^T \tag{6}$$

Random, $\vec{V}_5$: A normalized random vector. Depending on this vector's coefficient (w_5), an unpredictability is introduced to the swarmettes' flight.

3 Evolution of a Creative Swarm

The basic functionality presented in the previous chapter allows a swarm to create three-dimensional structures. Specific building tasks can be evolved by supplying a 3D structure towards which the swarm is supposed to orientate its construction.

3.1 Representation

The weights w_0 to w_5 and maximum values for acceleration A_{max} and velocity V_{max} result in the flocking behaviour of a swarm. A set of m rules, with $0 \leq m \leq 20$, determines the constructional abilities of a swarm. Each rule r_i of length $0 \leq n \leq 5$ has the form:

$$r_i = c_{i0} \wedge c_{i1} \wedge \ldots \wedge c_{in} \rightarrow a_i, \tag{7}$$

where a condition c_{ij} is fulfilled, if a construction element is found at a location $\vec{p}_{ij}$ from the construction element the swarmette has collided with. The rule consequence a_i consists of a specific action (one of Table 2) and a 3D vector as the action's parameter. Once a collision occurs, each rule r_i is tested and, if applicable its consequent action is executed. If more than one rule applies, the corresponding actions are executed in the order of their genotypical appearance. In total a swarm genotype g_s is described as the set of available alleles:

$$g_s = \{w_0, \ldots, w_5, r_0, \ldots, r_m\}. \tag{8}$$

3.2 Genetic Operators

Randomly generated two point crossover masks are the default choice for the recombination of the swarm genotypes. If there exist any dependencies within the agent's genotype (e. g., an instructional rule that makes only sense with another one within the same set), its partitioning into three parts very likely conserves them.

The offspring's number of rules is limited to the smallest order of the ancestors' rule sets. Hence, it may happen that the average size of the swarms' rule sets decreases in the course of evolution. A small set of rules and still good performance corresponds with Occam's razor [mit97]. Figure 2 illustrates the crossover routine. For two arbitrary genotypes g_i and g_j a crossover mask c_{ij} of the shorter genotype's length is generated. g' inherits an allele whose entry in c_{ij} is 0 from g_i, if it is 1 from g_j (vice versa for g''). In this example, both offspring have only 8 instructional rules. g_j's surplus of rules (17 instead of 8) is not considered.

$$g_i = \{w_0^i, \ldots, w_5^i, r_0^i, \ldots, r_8^i\}$$
$$g_j = \{w_0^j, \ldots, w_5^j, r_0^j, \ldots, r_{17}^j\}$$
$$c_{ij} = \{000000\,1111\,00000\}$$
$$g' = \{w_0^i, \ldots, w_5^i, r_0^j, \ldots, r_3^j, r_4^i, \ldots, r_8^i\}$$
$$g'' = \{w_0^j, \ldots, w_5^j, r_0^i, \ldots, r_3^i, r_4^j, \ldots, r_8^j\}$$

Figure 2: The alleles of two arbitrary genotypes g_i and g_j are combined in accordance with the crossover mask c_{ij} in order to generate the new offspring g' and g''.

Mutation (see Algorithm 1) is applied on every allele of all genotypes (see Equation 8) of a new generation. Conditions, action and the action's parameter of each rule undergo the mutation process separately.

At first the mutation operator checks whether it should alter the given value (v) or not according to a mutation rate (mr). If the decision is made in favour of alteration, an update value (Δv), smaller than a given mutation distance (md), is chosen and added to or subtracted from the original value. Whenever the resulting values leave an interval (defined by a lower bound lb and an upper bound ub), they are trimmed to the next boundary. The mentioned boundaries ensure that the evolved parameters make sense and the mutation distance defines the procedure's maximum effect. Table 3 shows the simulation's default parameters.

The next generations' members are chosen by means of fitness proportionate selection.

Algorithm 1 Mutation Procedure

Returns: An unchanged or mutated value v
  Generate a random value $t \in [0, 1]$
  if $t > mr$ **then**
    Set Δv to a random value between $-md$ and md
    $v \leftarrow v + \Delta v$
    Return $min(max(v, lb), up)$
  else
    Return v
  end if

3.3 Guidance with a Given 3D Structure

Each run of a simulation that is guided by interactive evaluation builds upon a certain idea. One might, for instance, try to achieve a construction that reaches very high. During the course of evolution the breeder might run into an unforeseen though interesting structure that inspires his/her objective's notion. However, when starting the simulation, the supervisor must have in mind a reasonably explicit conception. If one is able to map some attributes of the imagined structure onto a three dimensional construction (such attributes can be height or a general shape, etc), the provision of orientation towards an object turns out to be beneficial. In this subsection we suggest a fitness function which is only supposed to guarantee the adoption of a given shape's general features.

Evolution parameters	
Population size	20 swarms
Number k of best genotypes	10
Crossover rate	0.4
Mutation Rate	
In general	0.2
On rule actions	0.1
Mutation Distance	
Flocking parameters	0.05
Rule conditions	0.2
Alleles' Boundaries	
Flocking parameters	$[0, 2]$
Rule vectors	The world size
Maximum number of rules	20
Maximum number of conditions	5

Table 3: Settings of the evolutionary process

In order to guide the search, the difference between the construction of a swarm and a pre-defined three dimensional structure is used as the fitness rating. To measure the difference of the two 3D objects the set of built construction elements is tested for intersection against the set of given cubes. We define the following measures:

Covering Volume, C: Represents the summed intersections of built and pre-defined construction elements.

Non Covering Volume, $\bar{C}$: Is the volume of the construction that does not intersect with the pre-defined structure.

Fitness Object Volume, F: Is the total volume of the pre-defined structure.

The fitness function for a swarm is then:

$$fitness_{swarm} = \frac{C}{F} - \frac{\bar{C}}{F} \tag{9}$$

This function reaches its maximum when $C = F$, and $\bar{C} = 0$ which is the case if the whole given structure is rebuilt by the swarm. To ensure that the construction of a swarm is as near to the given 3D objects as possible, any outgrowth is rated negatively by the second term of Equation 9. The more construction elements are built that do not contribute to the given structure's approximation, the lower is the fitness of the swarm. Instead of direct subtraction of the volume of the misplaced construction elements, a softer penalty is imposed. Consequently, the punishment is small as long as the swarm creates fewer construction elements than necessary to fill up the given structure. Additionally, this gives the swarm an impetus to construction in general. However, if the built structure grows rampantly, the penalty will decrease the fitness of a swarm.

3.4 Initial Settings

The maximum order of the rule set as well as the maximum length of a single rule are given. The higher the number of conditions, the lower is the chance that a rule complies with. In order to leave the rules easily applicable, the number of conditions is limited to five. The maximum number of rules is set to 20.

Each condition of a rule is randomly initialized to a vector of the set of basic directions D, introduced in Section 2. The same holds for the vector that is part of the consequence of the rule. The action itself can be any of those presented in Table 2, with equal probabilities for changing the swarm center, for the creation of a new or for the destruction of an old construction element.

Triggering the creative behaviour of a swarm depends on at least one template construction element. For each breeding experiment the information where to place these template elements is defined in an extra file.

The structure which guides the evolutionary process is also read from a file. The genotype of each swarm along with its achieved fitness are stored in a protocol file. The evolutionary process can be resumed, if it had to be stopped. The k fittest genotypes are saved automatically in a separate file.

4 Evolutionary Discovery of Swarms that Build 3D Structures

In this chapter some interesting structures built by rule-based swarms are presented. Furthermore a case-study is conducted. The case-study comprises the evolutionary development of a swarm, attributes of its created structure and characteristics of its flocking behaviour. Finally a few examples show, how an already evolved swarm is interactively guided by providing derivations of the originally given 3D structure.

4.1 First Example Structures

Many swarms have been evolved that create interesting structures. In Figures 3, 4, 5, 7 and 6 several experiments' underlying 3D structures along with the constructions of well-performing swarms are shown. Except for the swarms of Figures 4 and 5, which have both a set of 18 rules, all the underlying swarms have 10 instructional rules and make use of 4 or 5 conditions.

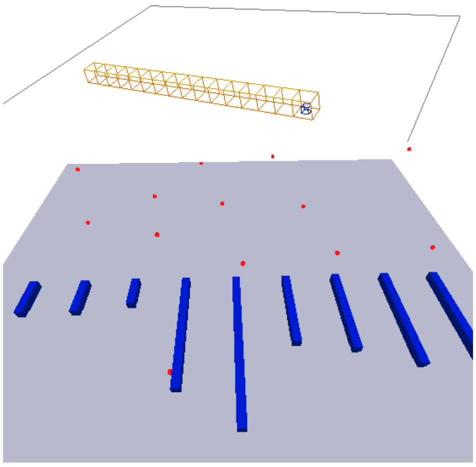

Figure 3: Top: The given 3D object that guides the evolutionary search. Bottom: An evolved swarm builds lines starting from any provided seed construction element (generation 15, fitness 0.0064).

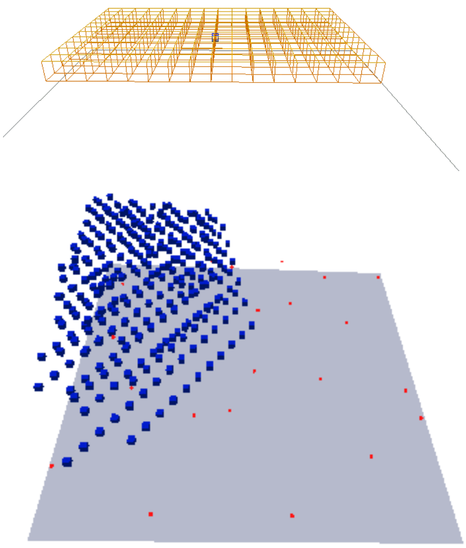

Figure 4: Top: Given are one plane and a template construction element at a small distance from the ground. Bottom: A fan-like structure is built (generation 506, fitness 0.0058).

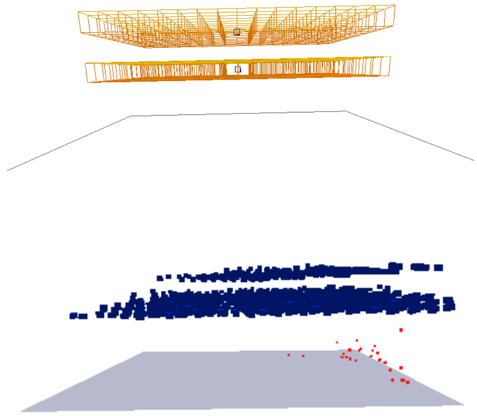

Figure 5: Top: Given are two planes. Bottom: A swarm of generation 314 creates "two-level flats" starting with two seed blocks at the corresponding heights (fitness 0.0122).

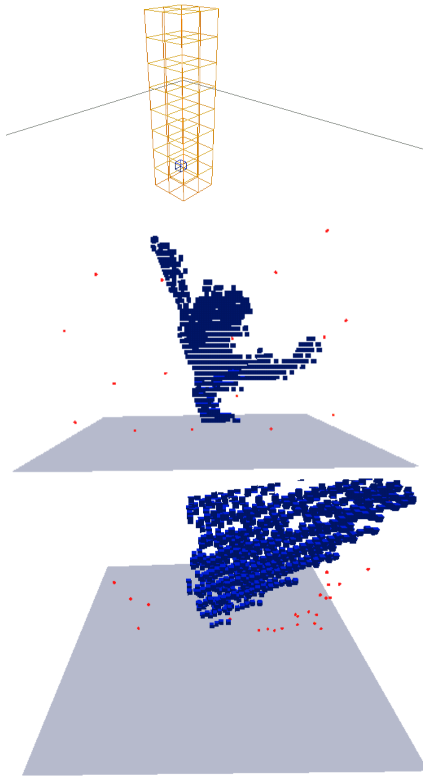

Figure 6: Top: A simple tower to guide the evolutionary search yields: An interesting shape, reminding of a statue (35th generation, 0.1970 fitness), the image in the middle. Bottom: A swarm that is divided into two flocks at an early stage. Both flocks loop back and forth from their sides to the construction. The many holes in the structure make it look like a bush (1143rd generation, 0.0066 fitness).

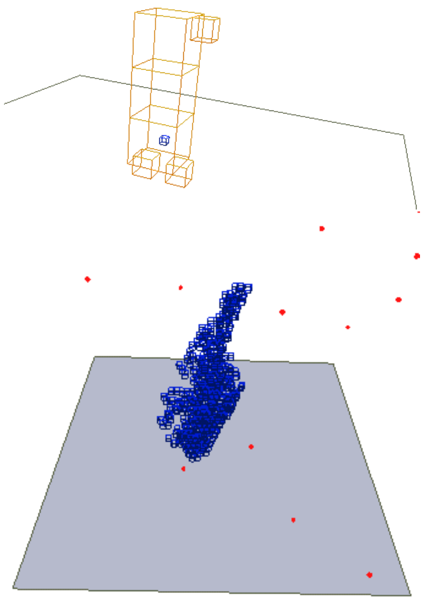

Figure 7: Top: An asymmetrical 3D structure to guide the search. Bottom: An asymmetric, organic looking shape (7th generation, fitness is 0.1305).

4.2 A Case-study: Approximation of a Tower

The given tower structure, displayed in Figure 6, has also been basis of the development of the swarm which is examined in this subsection. In general the swarm in question has similar attributes as the ones presented in Subsection 4.1: Its constructional behaviour is determined by a set of ten rules, whereas the maximum amount of conditions of these rules is five. It took 143 generations for the considered structure to occur and it achieves a fitness of 0.1926. The genotype of the swarm is presented in Tables 4 and 5. In order to increase the readability, we rounded the numbers. For the exact data, please refer to the appendix of [mam05].

Parameter	*Value*
Alignment	0.30
Separation	0.06
Cohesion	0.18
Random	0.00
Ground	0.17
Center	0.14

Table 4: The flocking parameters of the tower building swarm.

The overall behaviour of the swarm is intriguing. First it builds up a tower which is very close to the given structure (see Figure 8). Then it is fleeing from the world center to avoid further construction which might lead to fitness penalty (see Figure 9). The whole construction procedure takes approximately 200 simulated seconds, but the biggest part is already built after 80 seconds have passed. At the simulation start and as long as it takes to create the tower, the swarm stays around the center. Afterwards it is divided into three to four flocks which head as far away from the center as possible, lingering in the simulation world corners.

Rule conditions	*Action*
None	Recant swarm center
None	Destroy construction element at $(-0.7, 0.3, 0.1)^T$
$(-0.1, 0.6, -0.2)^T$	Set center to $(1.1, 0.7, -1.8)^T$
$(-0.1, 0.1, -1.6)^T$, $(-0.8, 0.6, 0.6)^T$, $(0.6, 0.2, -1.0)^T$, $(0.5, 0.2, -0.2)^T$	Set center to $(-0.8, 1.4, 1.1)^T$
$(-0.5, 1.3, 0.1)^T$, $(-0.3, 0.5, 0.4)^T$	Create element at $(-0.1, 0.3, -0.8)^T$
None	Create element at $(0.0, 0.2, -0.1)^T$
$(-0.4, 0.0, -0.2)^T$	Create element at $(-0.3, 0.4, 0.5)^T$
None	Create element at $(0.0, 0.0, 0.2)^T$
None	Create element at $(0.1, 0.2, 0.0)^T$
$(0.0, 0.2, -0.1)^T$	Destroy the collision partner

Table 5: The construction rules of the tower building swarm.

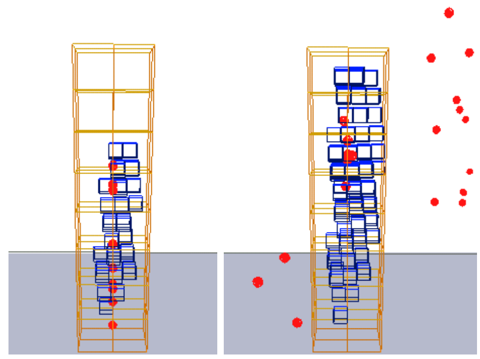

Figure 8: The images show intermediate states of the building process of a tower. The given structure consisting of the bigger cubes, elucidates the degree of the approximation of the swarm.

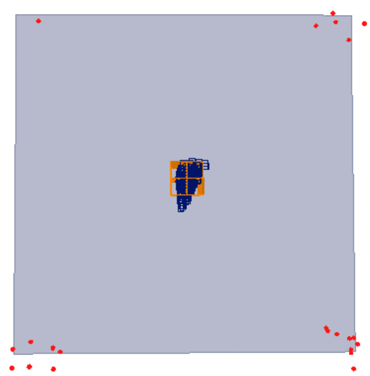

Figure 9: After the approximation of the tower the individuals flee into the simulation world corners (seen from above).

The construction of the swarm can be characterised as follows:

Compactness: The construction elements are built close to each other, therefore the construction is compact.

Structure: Although an iterative construction process can be observed, patterns or modules of construction elements are not apparent. However, the fact that the construction elements are built on top of each other to gain height and next to each other to approximate the given shape's breadth is sufficient to outline a structure.

Coordination: The swarmettes act in a coordinated manner. They build at different locations without endangering the total construction.

A closer look at the construction rules (Table 5) of the swarm helps to understand the change in its flocking behaviour. The succession of the rules plays an enormous role. Exempting the swarm from its center is only realized, if no subsequent rule defines the swarm center anew. Since the rules that redefine the swarm center ask for certain structural configurations around the agent to come into effect, the swarm is focused on the building as long as these conditions are fulfilled. Through steady alteration of the built structure it might occur that these conditions are not satisfied for agents at a certain location. As a consequence the intrinsic flocking behaviour of the swarm could come into action urging the swarm to the simulated world corners. This reasoning conforms to the seen phenomenon and is supported by the genotype of the swarm.

4.3 Guiding the Search with Diversified 3D Objects

Based on the rule-based tower building swarm, presented in Section 4.2, we further evolved variations of the original tower. Three different ideas for changing the tower shape are discussed (shown in Figure 10): its height extension, an additional branching "crown" and stairs on top of the original tower.

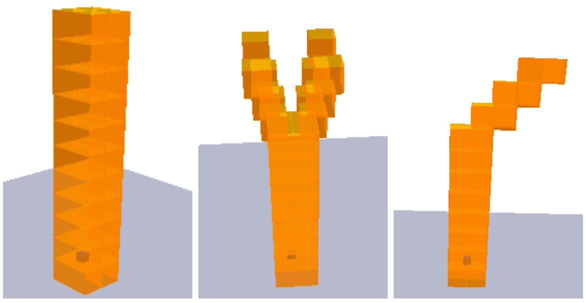

Figure 10: From left to right: A taller tower, an additional branching "crown" and stairs on top.

It took 184 generations until the tower building swarm has adopted to the elongated shape of the fitness structure. In general the same course of events takes place as seen in Figures 8: Now the agents build a somewhat higher tower before they flee from the world center.

Other results of the tower extension experiment are displayed in Figure 11. The images represent the results of two separate evolutionary runs, both starting with the genotype of the tower building swarm as discussed above. It is obvious that there is a general tendency of the swarms to fulfill the new requirements. The extension of the original tower's shape gives a strong impetus to build higher. The evolutionary process accomodates the shape's variance and thereby generalises beyond the new pre-defined structure.

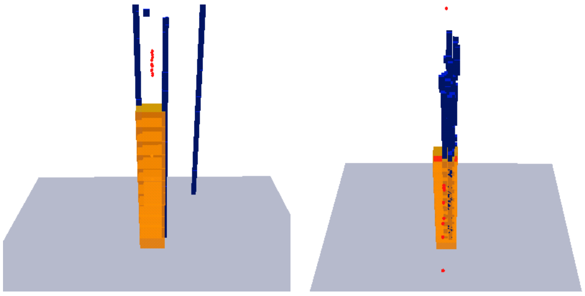

Figure 11: Left: Pillars arise after 1000 generations. Right: Extension of the originally provided 3D structure results in endless efforts to gain height (644th generation). Both images show the vertical line flight formation that contributes to the construction of the swarm.

After 1341 generations the original tower building swarm has adjusted to the new challenge: Now it contributes to the branching structure by extending its construction diameter with growing height (Figure 12).

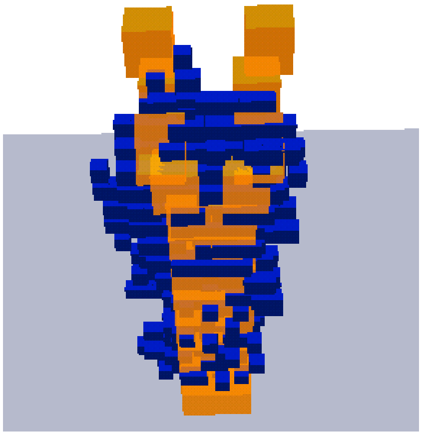

Figure 12: The branching crown on top of the tower, has been approximated within 1341 generations.

Figure 13 shows the approximation of the third shape variation. It seems to be hard to cope with the stairs on top of the tower. However, after only two generations the hillclimber strategy (with the tower building swarm as basis) yields a genotype with a fitness of 0.2087. A structure bent into the direction of the stairs is created by the further developed swarm.

A simulation run with no specific starting point has not been able to achieve equally good results within 1000 generations. Its best result has occurred in generation 925 with a slight tendency of building towards the stairs and a fitness of 0.0401.

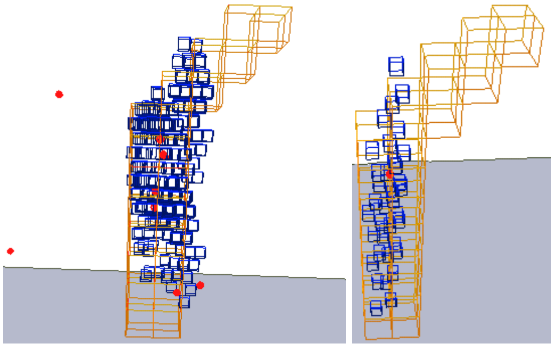

Figure 13: Left: The result of an evolutionary run with an optimized tower building swarm as basis. Right: An outcome without a specific initialization (925th generation).

5 Conclusion and Future Work

Normally the evolutionary search can be guided by the supervisor with simple selection or rating of a current population's phenotypes. A method is introduced that allows to direct the course of evolution by iteratively changing its objective.

Adjustments of the swarm model have to be made, too. Currently, the agent's genotype is limited to maximally five conditions per rule. So far artificial evolution has originated mostly swarms that obtain rules with at most four to five conditions, which clearly exhausts the given limit of conditions. Therefore the effect of an increased number of allowed conditions should be analysed.

Each of the presented swarms has one genotype which holds for all its agents. It has been shown that the presented swarm model works in general and yields some interesting results. However, more complex structures might arise by assigning individual genotypes to each swarm agent. The necessary symbiosis of genetically different individuals could be induced by a very careful use of coevolution.

Bibliography

[bon99] E. Bonabeau et al. (1999) ”Swarm Intelligence: From Natural to Artificial Systems”, Oxford University Press, New York, Oxford.

[bur04] I. Burleigh (2004) ”An Agent-based Model of the Lactose Operon (A Journey to the Centre of the Cell)”, Master's Thesis, University of Calgary, Canada, http://pages.cpsc.ucalgary.ca/~burleigh/LacOperon/.

[daw87] R. Dawkins (1987) ”The Blind Watchmaker”, Longman Scientific and Technical, Harlow.

[gar92] H. de Garis (1992) ”Artificial Embryology : The Genetic Programming of an Artificial Embryo”, Ch. 14 in: ”Dynamic, Genetic, and Chaotic Programming”, ed. Branko Soucek and the IRIS Group, Wiley.

[jac01] C. Jacob (2001) ”Illustrating Evolutionary Computation with Mathematica”, Morgan Kaufmann, San Francisco.

[jac03] H. Kwong and C. Jacob (2003) ”Evolutionary Exploration of Dynamic Swarm Behaviour”, IEEE Congress on Evolutionary Computation, Canberra, Australia.

[kwo03] H. Kwong (2003) ”Evolutionary Design of Implicit Surfaces and Swarm Dynamics”, Master's Thesis, University of Calgary, Canada.

[mam05] S. v. Mammen (2005) ”Evolving Swarms that Build 3D Structures”, Student Thesis, University of Erlangen-Nuremberg, Germany, http://www2.informatik.uni-erlangen.de/Lehre/SA-DA/download/Mammen.pdf?language=de.

[mit97] T. M. Mitchell (1997) ”Machine Learning”, McGraw Hill, Boston, Massachusettes.

[pil04] M. L. Pilat (2004) ”Wasp-Inspired Construction Algorithms”, University of Calgary, Canada, http://www. pilat.org/.

[rey87] C. W. Reynolds (1987) ”Flocks, Herds, and Schools: A Distributed Behavioral Model”, Computer Graphics, 21(4) (SIGGRAPH '87 Conference Proceedings) pages 25-34.

[sim91] K. Sims (1991) ”Artificial Evolution for Computer Graphics”, Proceedings of the 18th annual conference on Computer graphics and interactive techniques, pages 319-328, ACM Press, New York, USA.

[tho02] D. Thomas (2002) ”Aesthetic Selection of Developmental Art Forms”, Artificial Life VIII, The 8th International Conference on the Simulation and Synthesis of Living Systems, Sydney, Australia, pages 157-163, MIT Press, Cambridge.

[whi01] T. Whitelaw (2001) ”Breeding Aesthetic Objects: Art and Artificial Evolution”, Creative evolutionary systems, Section: Evolutionary creativity, pages 129-145, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.