Evolutionary Swarm Design of Architectural Idea Models

Sebastian von Mammen
Department of Computer Science
University of Calgary
2500 University Drive NW
T2N 1N4 Calgary, Alberta, Canada
s.vonmammen@ucalgary.ca

Christian Jacob
Department of Computer Science
University of Calgary
2500 University Drive NW
T2N 1N4 Calgary, Alberta, Canada
cjacob@ucalgary.ca

ABSTRACT

In this paper we present a *swarm grammar system* that makes use of bio-inspired mechanisms of *reproduction*, *communication* and *construction* in order to build three-dimensional structures. Ultimately, the created structures serve as idea models that lend themselves to inspirations for *architectural designs*.

Appealing design requires structural complexity. In order to computationally evolve swarm grammar configurations that yield interesting architectural models, we observe their *productivity*, *coordination*, *efficiency*, and their unfolding *diversity* during the simulations. In particular, we measure the numbers of collaborators in each swarm individual's neighborhood, and we count the types of expressed swarm agents and built construction elements. At the end of the simulation the computation time is saved and the created structures are rated with respect to their approximation of *pre-defined shapes*. These ratings are incorporated into the fitness function of a genetic algorithm. We show that the conducted measurements are useful to direct an evolutionary search towards interesting yet well-constrained architectural idea models.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*; I.2.6 [**Learning**]: Induction; I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems, coherence and coordination; J.5 [**Arts and Humanities**]: Architecture; J.6 [**Computer-aided Engineering**]: Computer-aided design (CAD)

General Terms

Algorithms, Design

Keywords

Swarm grammar, constructive swarm, generative representation, swarm model, stigmergy, boids, complexity

1. INTRODUCTION

Discoveries of intricate construction technologies applied by ancient cultures are usually met with great surprise [9]. It is generally assumed that architectural freedom evolves with scientific and technological progress [5]. Accordingly, the means to realize bold architectures have steadily grown [4]. Architects embrace the newly gained freedom — to implement almost anything conceivable — to devise innovative and compelling designs [5]. Consequently, the search for intriguing signature design ideas has shifted to the forefront of architectural work.

Inspired by building processes of social insects [7, 2] we have developed a computer-based, evolutionary swarm system to create 3D structures that lend themselves to architectural idea models [13]. Traditionally these models are the first three-dimensional realization of an architectural idea, still omitting details of its actual construction.

We have organized our paper as follows. We place our work in context in Section 2. In Section 3 we describe the swarm model that we have developed. In order to find swarm configurations that yield interesting architectural idea models we apply evolutionary computation. The interplay of genotype representation, phenotype simulation, their evaluation and deployed evolutionary operators is described in Section 4. Section 5 presents results of our evolutionary runs, followed by a summary and an outlook on future work.

2. RELATED WORK

Wasp nests, ant galleries and termite mounds are examples of the construction abilities of natural swarms [7]. Local neighborhood information, consisting of flock mates and environmental stimuli (*templates*), trigger the individuals' behaviors. Step by step, intricate architectural solutions emerge while individuals transport construction elements, place pheromones and react to traces left by their mates. This *stigmergic* approach of nest constructions of social insects has been reproduced in computational simulations [2, 19]. Refined simulations showed that even the consideration of physical constraints (like wind) do not negatively affect the modeled decentralized swarm construction of termites [15].

Abstract constructive swarm models are utilized to create traditional art [3], to craft virtual artistic sculptures [11], and for interactive art performances [18, 10]. Interactively trained rule-based lattice swarms have been used to reproduce human-like architectural construction [28, 29].

While interactive evolution has proven most adequate for breeding aesthetically pleasing art works (e.g. [23, 24, 11]), architectural designs quickly increase in complexity, thereby challenging the breeder [1]. In a semi-interactive evolutionary system the fitness of an individual is evaluated computationally as well as manually by a breeder. With the computational means to determine and compare the complexity in architectural constructions, aesthetics could remain the sole role of the breeder [17, 16].

Complexity can incorporate different aspects, such as: ecological diversity, complexity of construction (functionality), or internal complexity (also logical depth, e.g. hierarchical complexity) [22]. Different complexity measurements were tested on grammatical programs, respectively tree-like data structures, whose interpretation leads to the creation of virtual artifacts [8]. Under the assumption that a sound complexity measure scales with the size of the problem, it is suggested that the consideration of *modularity*, *reuse* and *hierarchy* yields reliable values of complexity.

However, these characteristics cannot be easily identified in constructive swarm systems which do not offer a one-to-one mapping from an encoding to the resulting artifact, unlike other generative representations such as L-systems [20]. Swarms are non-determinstic systems in which local interactions take place in parallel and create unforeseen interdependencies. Therefore, we need to analyze the swarm configuration, but we especially have to observe the resulting construction process to estimate the unfolding system complexity. For this purpose we measure the average number of flock mates within each individual's neighborhood, similar to the analysis of complex networks [12, 6]. We also observe the diversity of the expressed swarms (number of swarm agent types) and consider the number of types of construction elements used in the construction process. Furthermore, we prevent an outgrowth of (computational) complexity by killing off inefficient swarms that do not terminate within a fixed time-frame. To keep the construction of a swarm within well-defined boundaries and to meet architecturally expected proportions, the approximation of predefined shapes is rewarded with an increase in evolutionary fitness [26].

3. SWARM GRAMMAR MODEL

Swarm grammars [25, 11, 27] are a very expressive artificial swarm model. They merge the interaction dynamics of boid, i.e. agent-based, swarms [21] with the reproduction abilities of a generative grammatical system, like L-systems [20]. Hence, each swarm agent follows a set of *flocking urges*, e.g. *alignment* and *separation*, to constantly adjust its acceleration in accordance with its local neighborhood (Figure 10), while a grammatical production system determines the individual's transformation over time. Thus, a swarm grammar system comprises (1) a set of agent configurations, and (2) a set of production rules. Additionally, most swarm grammar models incorporate their individuals' ability to build structures by leaving construction elements in virtual space.

In preceding swarm grammar models the agents leave behind steady traces of construction elements in space [25, 11]. As a result, many emerging structures branch according to the reproduction of the swarm agents, resulting in plant-like, organic appearances. In a virtual creative system achieving an abundance of construction elements is inexpensive. We learn from social insect swarms that when stigmergic interplay directs the collective construction efforts, sophisticated and robust buildings can emerge (see Section 2). Thus, we have extended previous swarm grammar systems by event-based construction and reproduction rules, which we now describe in more detail.

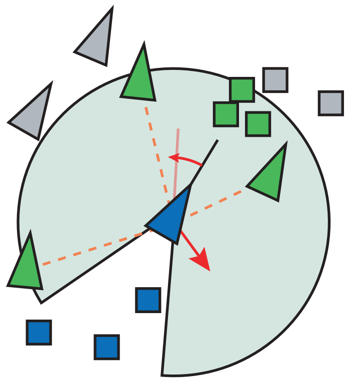

Figure 1: The green agents (polygons) and construction elements (boxes) are within the neighborhood perception of the blue swarm agent. This agent is urged to *align* with the perceived agents' orientations (upper arrow) and to *separate* from its flock mates (lower arrow) at the same time.

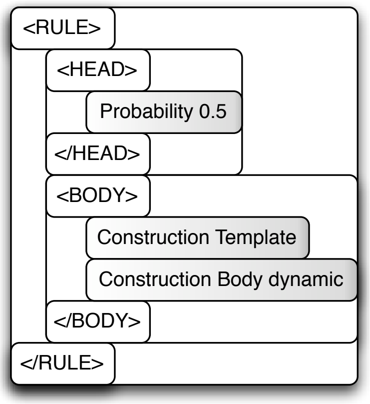

Figure 2: The syntax of a behavioral rule, exemplary for the encoding of the entire genotype.

The activation of a rule can be triggered by *timers*, the *perception of a specific construction element* or a *pheromone*, or plain *chance*. Empty rule heads result in the unconditional application of the rule body, whereas several conditions are interpreted conjunctively. Correspondingly, all directives listed in a rule's body are executed successively. The swarm agent can change its focus (*world center* [21, 14]) to a nearby agent, construction element or template. It can apply a grammatical substitution, thereby reproduce itself, differentiate into one or several different agent types, or die out. Third, the agent can place a construction element or a template in space. Templates, like pheromones, disappear after a certain time and do not contribute to the outcome of the construction but help to coordinate the construction process. In our setup templates are evaluated *qualitatively* by the agents: their mere existence can influence an agent's behavior [2]. For construction, we provide the three basic elements that are common in architecture: *rods*, *bodies* and *layers* [13]. Figure 2 depicts the encoding of a rule taken from an evolved swarm agent presented in Section 5: With a probability $p = 0.5$ the agent places a template and a cubic body construction element in space at each time step[1]. In the following section we will reveal more details about an agent's genotype.

4. EVOLUTIONARY SETUP

First, we describe how swarm grammars are encoded and modified during the evolutionary process. Second, we explain the process of fitness evaluation that directs the evolutionary search for architectural idea models.

4.1 Genotype and GA

In our breeding experiments we evolved populations of 20 swarm grammars over at least 30 generations. Each swarm grammar comprises 5 swarm agent types which are described by their flocking parameters [21, 14] and by sets of at most 10 behavioral rules as described in Section 3. As genetic operators we apply *fitness proportionate selection*, *elitism*, *mutation* and *crossover*. How the two latter operate on the swarm grammar genotypes is explained in the following paragraphs. The genotypes are encoded in tagged lists of key-value pairs, as illustrated in Figure 2.

Only one fourth of the next generation is subjected to mutation which is applied with a 50% chance to each gene. Numerical values are changed in accordance with a normally distributed maximal step size of 0.2. Hereby, the following intervals are considered: The flocking weights for *cohesion, alignment, separation*, for the *world center* urge and for the *random* urge are normalized to values between 0.0 and 1.0. An agent's mobility is limited to the absolute values 15 for velocity and 30 for acceleration. Its perception extends at most to a 5.0 radians radius and reaches at most ten units. The separation urge impacts an agent's acceleration only if its neighbors are not further than 5.0 units [14]. On mutation, rule conditions and actions are equally likely generated anew, deleted, or inserted. In the generation of conditions and actions each available directive, e.g. 'Change focus' or 'Reproduce', are chosen with the same probability. If a directive requires a parameter, it is chosen randomly as well. For instance, if 'Construction' has been determined as new directive, 'Rod', 'Body', 'Layer' and 'Template' are equally likely chosen as its parameter.

$5/8th$ of the next generation of swarm grammars result from recombination of the parents. Each behavioral rule and the lists of flocking parameters that appear in the genotype of a swarm grammar are used for recombination. Here, too, we apply the operation on each considered gene with a 50% chance. An alternative crossover implementation considers only the agents of a swarm grammar for recombination. Elitism transfers the fittest eighth of the parents to the next generation.

Individuals that are not assigned a fitness value greater than zero are considered extinct. If the parent population is diminished, the genetic algorithm generates an equally reduced population of successors. However, the population is automatically filled up with newly generated swarm grammars. This mechanism counterbalances the negative influence the genetic operators can exercise due to incomputable rule sets.

4.2 Fitness Evaluation

At the beginning of a simulation all N swarm agents[2] of a swarm grammar are expressed and initialized around the center of the virtual space (up to 10 units in x and y direction). Close by, at the bottom center of the pre-defined shape, at coordinates $(5, 5, 0)^T$, a template appears hinting at an ideal spot for construction (Figure 3(a)). For a specified time period of $\Delta t = 8sec$ the swarm agents coordinate, build and reproduce. Then the construction process is stopped, all data is written into a file and the next swarm grammar is evaluated. The fitness is evaluated based on the goals to *limit computational and constructional outgrowth* and to *promote production, diversity* and *collaboration.* We will explore these constraints in more detail in the following sections and then propose a fitness function that incorporates these aspects.

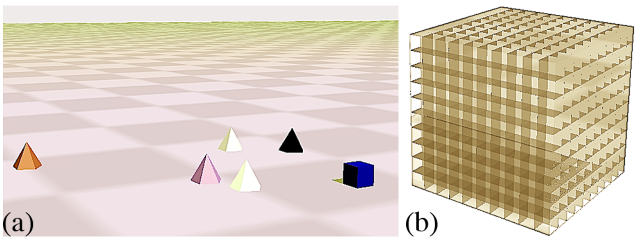

Figure 3: (a) Initial simulation state: 5 agents (polygons) are randomly placed in the vicinity of a template (cube). (b) Emerging structures are compared against this pre-defined shape consisting of 10^3 small cubes.

4.2.1 Preventing Escalation

Uncontrolled agent reproduction can quickly lead to an exponentially growing demand for computing resources. In order to avoid such an excess of resource usage, a simulation process taking longer than 100 real seconds is terminated and is not considered for further evolution. Additionally, the computing time t for a swarm grammar is stored as a variable in order to determine a specimen's fitness. On the one hand a certain degree of complexity in the emerging structures is desirable. On the other hand an outgrowth of (computational) complexity has to be avoided.

The same idea is realized when the swarm construction is compared against a pre-defined shape at the end of the simulation: constructions within a certain range of the target template are rewarded, whereas outgrowing the pre-defined limits is unproductive. Hereby, as in [26], the pre-defined shape consists of smaller cubes with edge size 1.0 (Figure 3(b)). We determine the ratio r_p between the number of these cubes that are penetrated by construction elements versus the total number of cubes comprised by the predefined shape.

---

[1]The keyword *dynamic* that occurs in the rule in Figure 2 means that the body construction is rotated according to the agent's orientation.

[2]For our experiments N=5.

Adding the ratio r_p to a swam grammar's fitness value rewards the swarm's productivity but only within certain boundaries. From a different perspective, it promotes constructions that retrace the provided pre-defined shape. Independently of the pre-defined shape, the total number of placed construction elements n_c can be utilized to further assess productivity and to limit the extent of construction as well.

4.2.2 Promoting Diversity

Since the construction patterns of individual swarm agents may vary, a wide diversity in constructions can be expected, that are built by a large number of different swarm agents. Even a homogeneous set of swarm agents can achieve greater diversity than a single swarm individual, as (1) the agents can influence each other's behavior, and (2) the same construction processes can be conducted in parallel. We express these observations numerically by r_a, the ratio of active agent types during a simulation to the total number of available agent genotypes, and by n_a, the total number of expressed agents.

Whenever different types of construction elements (rods, layers, bodies) are employed, an increase in structural diversity can be expected. As a consequence, the ratio r_c of employed construction elements to the number of available types is also considered for fitness computation.

4.2.3 Fostering Collaboration

As an alternative to the deployment of construction elements, swarm agents may drop templates that do not contribute to the construction and last for a short period of time only (for 20 iterations in the presented simulations). Consequently, time-critical signals can be propagated through templates, thus promoting collaboration among the swarm agents. We therefore also measure the ratio r_t of created templates to those that actually trigger a behavioral rule.

Swarm interaction is based on each agent's awareness of other agents. Therefore, we compute the average ratio r_n of agents that see each other to the total number of agents. For larger r_n the swarm agents stick together, whereas smaller r_n values reveal a very loose flight pattern — both of these extreme situations render collaboration difficult. For instance, a swarm grammar with $r_n = 0.87$ might form a clump as seen in Figure 4(a), whereas $r_n = 0.08$ can be an indication for uncoordinated growth as seen in Figure 4(b).

4.2.4 Proposed Fitness Function

The factors explained above are taken into consideration by the following scalar fitness function for a swarm grammar, f_{SG}. The terms g_n, g_c and g_a transform the corresponding variables to normalized values between 0.0 and 1.0 according to their semantics: A neighborhood ratio not too close to 0.0 or 1.0 is presumably beneficial. Reasonable amounts of expressed agents and placed construction elements are contributing to the fitness as well, especially, if these efforts do not overly extend the computation time t. We therefore arrive at the following fitness function:

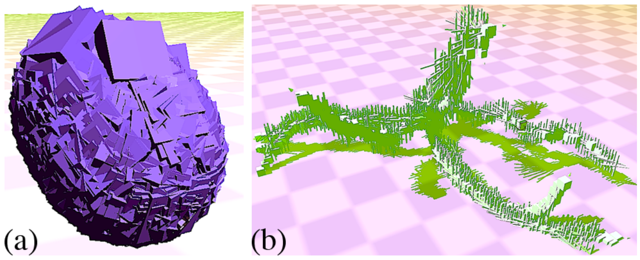

Figure 4: Neighborhood perception r_n during the construction process can sometimes be linked to the emerging structures. (a) A very compact structure emerges with $r_n = 0.87$. (b) Swarm agents drift away from each other, which yields a low perception rate $r_n = 0.08$.

$$
\begin{aligned}
f_{SG} &= r_p + r_a + r_c + r_t + g_n + \frac{g_c + g_a}{\sqrt{max(t,1)}} \\
g_n &= sin(\pi * r_n) \\
g_c &= sin(\pi * 0.005 * min(n_c, 200)) \\
g_a &= sin(\pi * 0.005 * min(n_a, 200))
\end{aligned}
$$

This fitness evaluation is used in our experiments, which we describe in the following section.

5. RESULTS

A successful search for architectural idea models heavily depends on the effectiveness of the genetic algorithm, especially on the crossover operator and on the fitness evaluation. Therefore, we first discuss our findings about the influence of the operator and fitness function on the resulting architectural constructions, before a variety of phenotypes is presented and analyzed.

5.1 Fitness Evolution and Crossover Points

Figure 5 depicts representative graphs of the fitness evolution in two independent experiments.

In the first experiment we apply a crossover operator $c1$ on rules and sets of flocking parameters only. In the second experiment each of the swarm grammars' N agents is considered for recombination (crossover operator $c2$). The average and the maximum fitness values of each generation are shown in the graphs avg_c1 and max_c1 in regards to $c1$, and in avg_c2 and max_c2 in regards to $c2$, respectively.

Elitism ensures that the best individuals are transfered unchanged into the next generation. Noise in the sequence of maximum fitness values (max_c1 and max_c2) is due to randomness in the simulations. As shown, max_c1 usually rises slower but does not differ much from max_c2. The development of average fitness values is of particular interest. The tendency of avg_c1 to stay considerably below avg_c2 is not a coincidence. If only agents of relatively successful swarm grammars are exchanged, the offspring's success mainly depends on the agent interaction encoded in their behavioral rules. Underachievement and thereby extinction can happen, but is less frequent than with recombination working on the building blocks of the agents' genotypes. Especially

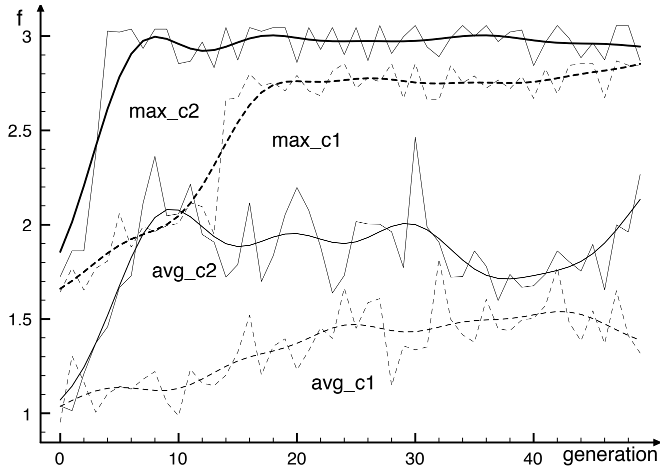

Figure 5: Fitness evolution in two experiments implementing different crossover operators. The upper graphs represent the maximal fitnesses achieved in each generation, whereas the lower graphs depict the average fitnesses of each generation.

the exchange of behavioral rules can lead to a swarm grammar's quick extinction. As soon as the agents reproduce themselves too frequently, the computing time rises and can easily exceed the maximum allowed timeframe. While the average fitness of the crossover on agents achieves a better development, the other crossover operator leads to a population of much greater diversity. On the one hand, the recombination possibilities are much greater when genetic information on the agent behavior level is considered. On the other hand, the high extinction rate allows new genotypes to enrich the gene pool.

In our experiments certain fitness properties were obtained faster than others. r_p, g_n, g_c and g_a had fast and great impact on f_{SG} and, consequently, on the evolutionary development. We were not able to promote rising values for r_a, and r_c which mostly exhibited erratic changes, or for r_t which did not contribute at all. Consequently, the following, simplified fitness function might have sufficed to breed the presented examples.

$$f_{SG}^{simple} = r_p + g_n + \frac{g_c + g_a}{\sqrt{max(t, 1)}}$$

Also, since the 'tasks' that correspond to the ineffective variables seem too difficult to be learned instantly, either partial task fulfillment (e.g. first, the placement of a template and second, the response) should receive a reward. Alternatively, the generation of behavioral rules could be constrained, thereby reducing the search space for 'useful' rules.

5.2 Architectural Designs

The outlined experimental setup results in a wide variety of architectural designs, a selection of which is presented in the following paragraphs. We differentiate between three structure categories depending on the actual construction elements: rod, body or layer. This classification schema concurs with actual architectural categories [13]. Additionally, we introduce a category for swirly architectural idea models.

The discussion of the examples underlines that the mapping from a swarm grammar genotype to the corresponding structure is not trivial. The provided characteristic measures drive the evolutionary process, yet one can hardly infer specific architectural categories from these measures, as we will demonstrate.

5.2.1 Rod Architectures

Figure 6 shows four examples of constructions in which rods dominate their visual character. In fact, the structure depicted in Fig. 6(c) only comprises about 60 rods, a mere 3% of the employed construction elements. The remaining three architectures, Fig. 6 (a), (b) and (d), however, are based on 50% to 60% rods. Investigation of the genotypes reveals that the rod-architecture swarms' behaviors are not synchronized through timer conditions. In general however, they exhibit rule conditions similar to those of the following examples (44% unconditional, 18% probabilistic, 15% on template sight, 14% on agent sight, and 10% timers). The presented swarm grammars' tracing success value r_p and neighborhood perception r_n are listed in Table 1.

The four phenotypes displayed in Figure 6 show a nice diversity. Fig. 6(a) exhibits three completely different segments, arranged from left to right. The first looks like a pile of sheets, the second like a spiky armor and the third does not only mix cubic construction elements and elongated rods, but also mixes two colors. The model in Fig. 6(b) can also be divided into three parts. From the bottom-left of the image a lattice tail loosely connects to the main part of the model. From there on, rods are laid out horizontally resembling stairs that lead to the top of an impenetrable spherical heap of rods. Fig. 6(c) shows a multifarious construction. Cubic elements are arranged at the bottom and the top. They are interconnected with a densely packed, dynamically shaped hose. Rods are floating in a wave-like fashion around the model's peak. Model Fig. 6(d) embodies the swarm dynamics of the construction process. The movements of the flocks of swarm agents create the impression of dynamic parts. This vivid impression is supported by the rough looking combination of layers and rod elements.

Model	r_p	r_n	Model	r_p	r_n
Fig. 6(a)	0.89	0.16	Fig. 8(a)	0.77	0.51
Fig. 6(b)	0.59	0.50	Fig. 8(b)	0.71	0.30
Fig. 6(c)	0.98	0.41	Fig. 8(c)	0.85	0.54
Fig. 6(d)	0.19	0.43	Fig. 8(d)	0.53	0.19
Fig. 7(a)	0.30	0.28	Fig. 9(a)	0.81	0.35
Fig. 7(b)	0.32	0.005	Fig. 9(b)	0.55	0.52
Fig. 7(c)	0.80	0.10	Fig. 9(c)	0.89	0.64
Fig. 7(d)	0.75	0.16			

Table 1: Characteristic values of the presented swarm grammar architectures.

5.2.2 Body Architectures

Figure 7 presents architectural idea models that are mainly assembled of (cubic) body construction elements. In fact, their share of all utilized construction elements varies be-

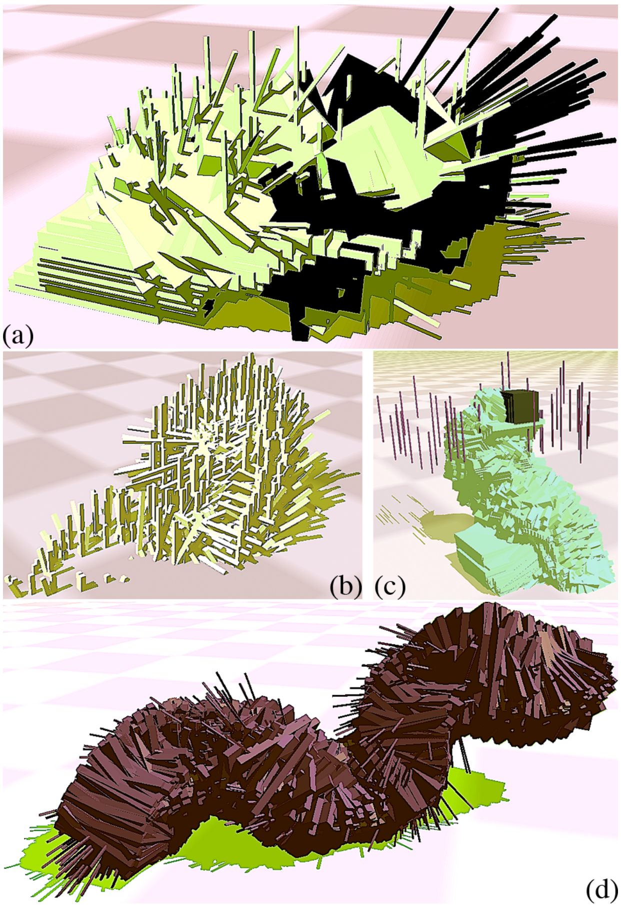

Figure 6: Rod-based architectural idea models.

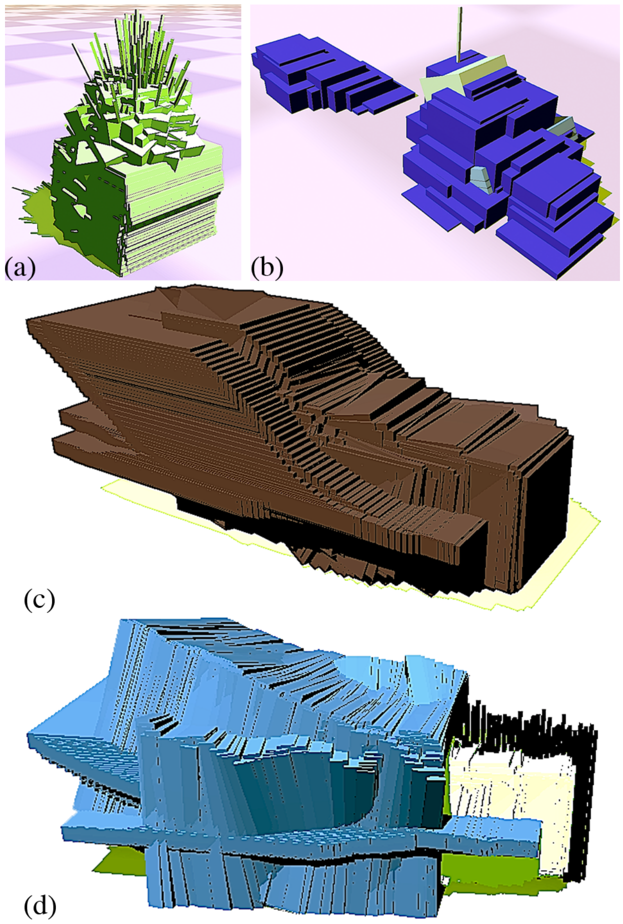

Figure 7: (Cubic) bodies coin the character of these architectural idea models.

tween 30% and 50%. Simple as it might appear, the construction in Fig. 7(a) achieves a very good approximation of the pre-defined shape (Table 1) and grows an extended set of bodies and rods on top of the bottom-up sequence of layered elements. The second example, Fig. 7(b), is distinct by its sparse use of different construction elements. An interplay of flocking swarm agents is obviously not required for the displayed model, as an extremely short perception radius of 0.6 units (maximally 10.0) keeps the swarm agents' neighborhood perception very low (Table 1). Fig. 7(c) presents a futuristic design that emerges through three interwoven construction mechanisms. (1) Cubic body elements form the main part of the model. (2) Layers flank the main part along the entire edge length. (3) Both layers and cubic body parts are rising in tandem to complete the construction with an elevated, inclined platform. The construction rule shown in Figure 2 belongs to an agent involved in the construction of Fig. 7(c). In fact, another rule makes the same agent differentiate upon sight of a body construction. The last instance of body-based architectures is shown in Fig. 7(d). Here, a dynamic character is introduced into the otherwise rather strict body architectures as seen in Fig. 7(a), (b) and (c).

5.2.3 *Layer Architectures*

Figure 8 displays four tower constructions that are coined by the employment of layer construction elements. In fact, Fig. 8(c) only utilizes 5 layers that can be spotted at the bent, the remaining 99.95% of the model consist of rods and body construction elements. Fig. 8(a) and (b) look very similar. Yet, they originated from completely independent experiments. Their characteristic values, too, resemble each other, except for the neighborhood ratio r_n (Table 1). Figures 8(a) and (b) consist of 25% and 17% layers, respectively. During both construction processes, agents transform/reproduce 15 times. Their visual resemblance is striking: From the bottom a rather rigid and straight stem is drawn upwards for about $3/4th$ of the total height. Then, body construction elements rise to a podium that is ornamented by several rods. During the construction of Fig. 8(d), agents reproduce 76 times. The increasing number of identical agents steadily widens the diameter of the construction (67% layers). The interplay of the swarms results in a rhythmic construction pattern that gains momentum towards the model's peak.

5.2.4 *Swirly Architectures*

Figure 9 presents three architectures that embody the actual swarm dynamics during the construction processes. In Fig. 9(a) a homogeneous set of five agents swirls around a rising path while dropping layers and rods. The resulting

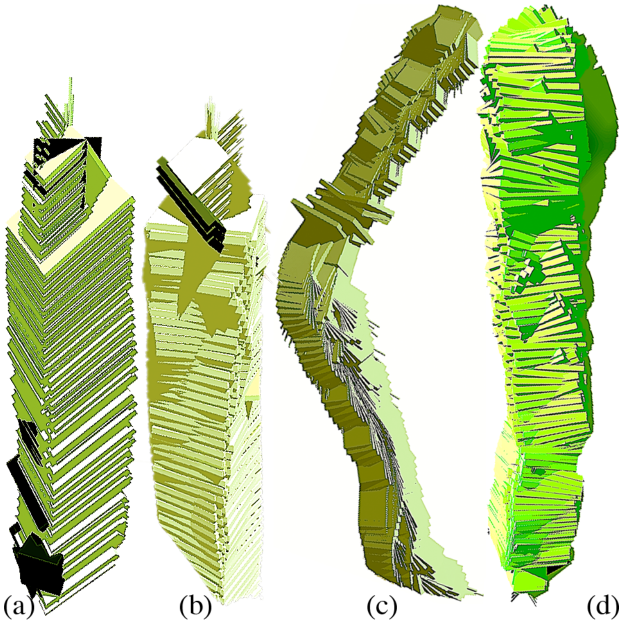

Figure 8: These tower architectures are mainly assembled of layer construction elements.

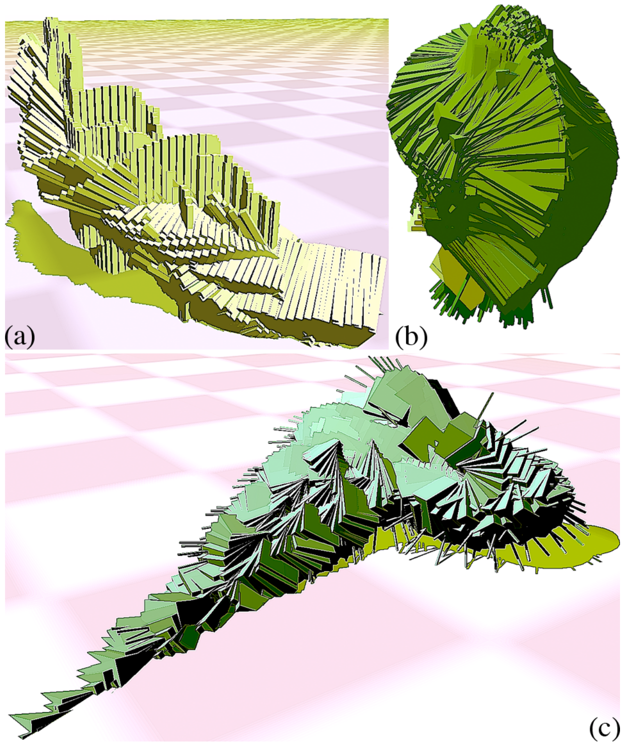

Figure 9: The displayed architectures emphasize the swarm dynamics of the construction process.

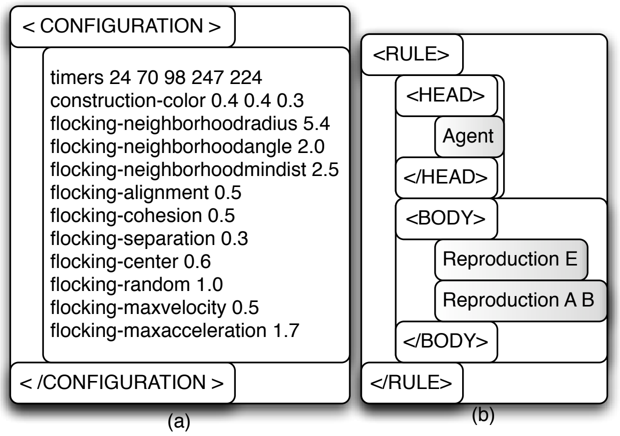

Figure 10: (a) The flocking parameters of the operative agent that wraps rods around the skeletal structure in Figure 6(b). (b) The spawning rule to delegate construction, employed in Figure 6(c).

construction receives good credit for the approximation of the pre-defined shape and proves that a relatively low neighborhood ratio $r_n = 0.35$ may very well lead to an intriguing, vivid swarm architecture (Table 1). The skeletal structure of Fig. 9(b) is assembled of rods and body construction elements. Several swarm agents wrap around and cement the inner construction with waves of rods. Crucial for this interplay is the probability-driven reproduction of the 'foremen' and their differentiation into mere operative swarm agents that do nothing but place construction elements. The emergence of a tight flocking pattern also strongly influenced the outcome. Figure 10(a) depicts the whole set of flocking parameters that determine the operative agent's flight. Cohesion and alignment are forces to keep the agents orderly together. When combined with a tendency for separation and randomness, the bulge formations can emerge. Fig. 9(c) displays a very complex swarm grammar: During the construction process agents spawn 725 times which might have led to the long computation time of $t = 63.3sec$. During the interplay of the expressed swarm agents, one of them is responsible for the reproduction and differentiation — the corresponding behavioral rule is displayed in Figure 10(b). One agent only places rods, another one only layers. The fourth involved agent places a rod, a body and a layer all at once but with a very low probability $p = 0.2$.

6. SUMMARY AND FUTURE WORK

We have presented an extended swarm grammar model that is capable of stigmergic construction of architectural idea models. In order to guide the evolutionary search we prevent structural and computational outgrowth by rewarding the approximation of a pre-defined shape and fast computation. Productivity, diversity and collaboration are furthered by counting events of construction, reproduction and by measuring neighborhood perception. Examples of successfully bred swarm architectures are presented and discussed. Although the introduced measures are efficient to guide the evolutionary search for innovative architectures, they cannot be directly linked to the architecture's properties.

For future work we consider the following steps. (1) Investigation of the temporal development of the perceived neighborhood ratio r_n, where a series of cyclic or jumping values might bear constructions different from those emerging based on constant values. (2) Providing incentives for stigmergic interdependencies to further collaboration. (3) Exploration of the impact of alternative pre-defined shapes (e.g. convex geometries) on the diversity and the design of emerging architectures. (4) Protocols of our evolutionary experiments underline the importance of an effective fitness function and effective genetic operators. Here, too, further investigation is necessary to find an optimum for diverse, appealing, and fit constructions that further facilitate the exploration of architectural idea spaces.

Evolutionary swarm design of architectural idea models works. However, in order to render this technology applicable for architects it has to be fitted according to their needs. Hereby, the main goals are the input of stronger constructional limitations as well as an interactive way to promote the development of compelling designs.

7. REFERENCES

[1] B. G. Bezirtzis, M. Lewis, and C. Christeson. Interactive evolution for industrial design. In *Conference on creativity & cognition*, pages 183–192, New York, NY, USA, 2007. ACM.

[2] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems.* Oxford University Press, New York, 1999.

[3] T. Csaszar. Swarm. *American Craft*, 66(4):44–49, 2006.

[4] K. Frampton. *Modern Architecture: A Critical History.* Thames and Hudson, 4 edition, 2007.

[5] J. Glancey. *Story of Architecture.* Dorling Kindersley, 2001.

[6] C. A. Hidalgo and A.-L. Barabasi. Scale-free networks. *Scholarpedia: The free peer reviewed encyclopedia*, 2006.

[7] B. Hölldobler and E. O. Wilson. *The Ants.* Springer-Verlag, Berlin-Heidelberg, 1990.

[8] G. S. Hornby. Measuring complexity by measuring structure and organization. In D. Srinivasan and L. Wang, editors, *2007 IEEE Congress on Evolutionary Computation*, pages 2017–2024, Singapore, 2007. IEEE Press.

[9] J.-P. Houdin. *Khufu: The Secrets Behind the Building of the Great Pyramid.* AUC Press, 2006.

[10] C. Jacob, G. Hushlak, J. E. Boyd, M. Sayles, and M. Pilat. Swarmart: Interactive art from swarm intelligence. *LEONARDO*, 40(3):248–254, 2007.

[11] C. Jacob and S. von Mammen. Swarm grammars: growing dynamic structures in 3d agent spaces. *Digital Creativity: Special issue on Computational Models of Creativity in the Arts*, 18, 2007.

[12] S. Kauffman. *At Home in the Universe: The Search for the Laws of Self-Organization and Complexity.* Oxford University Press, 1995.

[13] W. Knoll and M. Hechinger. *Architektur-Modelle: Anregungen zu Ihrem Bau.* Deutsche Verlangs-Anstalt, Munich, Germany, 2006.

[14] H. Kwong and C. Jacob. Evolutionary exploration of dynamic swarm behaviour. In *Congress on Evolutionary Computation*, Canberra, Australia, 2003. IEEE Press.

[15] D. Ladley and S. Bullock. Logistic constraints on 3d termite construction. In M. Dorigo, M. Birattari, L. M. Blum, F. Mondada, and T. Stutzle, editors, *Fourth International Workshop on Ant Colony*, pages 178–189. Springer, Berlin, 2004.

[16] P. Machado, J. Romero, and B. Manaris. Experiments in computational aesthetics. In P. Machado and J. Romero, editors, *The Art of Artificial Evolution*, Natural Computing Series. Springer, 2007.

[17] J. McCormack. Facing the future: Evolutionary possibilities for human-machine creativity. In P. Machado and J. Romero, editors, *The Art of Artificial Evolution*, Natural Computing Series. Springer, 2007.

[18] J. Nembrini, N. Reeves, E. Poncet, A. Martinoli, and A. Winfield. Mascarillons: flying swarm intelligence for architectural research. In *Swarm Intelligence Symposium 2005*, pages 225–232. IEEE Press, 2005.

[19] M. Pilat. Wasp-inspired construction algortihms. Technical report, University of Calgary, 2004.

[20] P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants.* Springer-Verlag, 1996.

[21] C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *SIGGRAPH '87 Conference Proceedings*, volume 4, pages 25–34, 1987.

[22] P. Schuster. How does complexity arise in evolution. *Complex.*, 2(1):22–30, 1996.

[23] K. Sims. Artificial evolution for computer graphics. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, volume 25, pages 319–328, New York, 1991. ACM Press.

[24] D. Thomas. Aesthetic selection of developmental art forms. In *Artificial Life VIII, The 8th International Conference on the Simulation and Synthesis of Living Systems*, pages 157–163, Cambridge, 2002. MIT Press.

[25] S. von Mammen and C. Jacob. Genetic swarm grammar programming: Ecological breeding like a gardener. In D. Srinivasan and L. Wang, editors, *2007 IEEE Congress on Evolutionary Computation*, IEEE Press, pages 851–858, 2007.

[26] S. von Mammen, C. Jacob, and G. Kokai. Evolving swarms that build 3d structures. In *2005 IEEE Congress on Evolutionary Computation*, volume 2, pages 1434–1441. IEEE Press, 2005.

[27] S. von Mammen, C. Jacob, and J. Wong. Virtual constructive swarms: Compositions, inspirations, and designs. In *Applications of Evolutionary Computing, Proceedings of EvoWorkshops 2008*, Berlin-Heidelberg, 2008. Springer-Verlag.

[28] Y. Zeng, P. B. Dennis, and C. H. Jorge. Multiagent based construction for human-like architecture. In *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–3, New York, NY, USA, 2007. ACM.

[29] Y. Zeng, C. H. Jorge, and P. B. Dennis. Swarmarchitect: a swarm framework for collaborative construction. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 186–186, New York, NY, USA, 2007. ACM.