The Design of Complex Architectures by Simple Agents

Eric Bonabeau
Guy Theraulaz
François Cogne

SFI WORKING PAPER: 1998-01-005

THE DESIGN OF COMPLEX ARCHITECTURES BY SIMPLE AGENTS

Eric Bonabeau*, Guy Theraulaz and François Cogne*****

** Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501, USA*
e-mail: bonabeau@santafe.edu
*** Laboratoire d'Ethologie et de Psychologie Animale, CNRS - UMR 5550, Université Paul Sabatier, 118 route de Narbonne, 31062 Toulouse Cédex, France*
e-mail: theraula@cict.fr
**** France Télécom CNET, Centre Lannion B - RIO / TNT, 22307 Lannion Cédex, France*

ABSTRACT

Complex architectures grown by simple agents moving randomly on a 3D lattice and depositing bricks deterministically depending on local configurations of bricks are presented. Some of these architectures are strikingly similar to real wasp nest architectures. But most algorithms, in the space of all possible algorithms of the kind described above, will produce structureless shapes. The space of possible architectures grown by artificial agents on a lattice, and capable of depositing two types of bricks according to the local state of the environment, is huge and long to explore, even if one restricts one's attention to (at least partially) spatially isotropic rules. To overcome this difficulty, a genetic algorithm (GA) has been used with a heuristic fitness function to explore the space of possible architectures. The fitness criterion is based on a simple observation: in algorithms that generate coherent architectures, many micro-rules are used, whereas in algorithms that generate structureless shapes, only one rule or a few rules are actually used in the course of the simulation. Results of this study are reported and discussed.

1. INTRODUCTION

A lot of animals can produce very complex architectures which fulfill numerous functional and adaptive requirements (protection from predators, substrate of social life and reproductive activities, thermal regulation, etc.) [5,8,9,12,13,14,16,17,18,22,23]. Among them, social insects are capable of generating amazingly complex functional patterns in space and time, although they have limited individual abilities. Deneubourg [3] showed that chemical cues could organize part of the building activity of termites through a process of self-organization. Self-organization [4,19] is thus a first option: in short, self-organization takes place in many-body systems when a critical threshold is reached. For example, there must be a critical number of individuals for the emergence of pillars in the case of termites: in effect, a termite picks up a soil pellet, impregnates it with an oral secretion which diffuses once the pellet is deposited; this diffusing substance, called cement pheromone, attracts other termites and stimulates them into depositing more pellets in the vicinity of the initial pellet; because the cement pheromone decays, a minimal number of termites is required to maintain the attractiveness of pellet deposits, which may then become pillars. Another option, complementary to, and possibly compatible with, self-organization, is stigmergy, a term introduced by Grassé [10,11] to describe the indirect communications taking place among individuals through the dynamically evolving shape in construction. Stigmergy was originally defined by Grassé in his pioneering studies on the reconstruction of the termite nest of *Bellicositermes natalensis* [10]. In order to explain the coordination of individuals' tasks, Grassé showed that the regulation of the building activity does not depend on the workers themselves but is mainly achieved by the nest structure: a stimulating configuration triggers a response, a building action, by a termite worker, transforming the configuration into another configuration that may trigger in turn another (possibly different) action performed by the same termite or any other worker in the colony. But Grassé did not show how this mechanism could canalize building and lead to specific structures.

Because it is complicated, experimentally, to track individual behaviors in detail over long periods of time — every individual has to be labeled and recorded, but even in this case it may be hard to determine what individuals are actually doing, because it requires detecting behavioral regularities in a noisy background —, one solution to model the building behavior of social insects is to use some coarse-grained level of description, where individuals behave, say, as automata responding to specific stimuli. Eventhough we have to proceed cautiously when drawing a parallel between automata and animals, such a description may not be too far from biological reality in some cases: for example, it has been observed that wasps may have a very clear if-then/yes-no behavior [6,7]. In what follows, we will be implicitly referring to wasps and wasp nests, although our results could in principle be separated from any underlying biological substrate. The stimuli which the wasps are sensitive to may be of a chemical nature, or may simply correspond to shapes, i.e., particular configurations of matter, that can be perceived in various ways. For instance, it is instructive to observe a wasp lengthen a cell: she senses height differences between the different parts of the cell, and then adds matter where it is necessary [15].

We undertook a modeling approach [1] by looking for the simplest automata that could generate complex architectures similar to those observed in nature. We naturally started with algorithms that could hardly be made simpler: our automata are agents moving in a 3D space, that behave locally in space and time on a pure stimulus-response (SR) basis. The deposit of an elementary building block (hereafter called a brick) by an agent just depends on the local configuration of bricks in the cells surrounding the cell occupied by the agent. A random exploration of the space of such algorithms first yielded no interesting result, i.e., only random shapes. Then, by working "backwards" from the shapes to be generated to the potentially corresponding algorithms, we eventually found it possible to produce complex shapes, some of them strikingly similar to those observed in nature, with algorithms belonging to this simple family [1]. Moreover, we also showed [20,21] (1) the smoothness of the mapping from algorithms to architectures (i.e., unless one is in the vicinity of a singularity, two close algorithms generate two close architectures), and (2) the compactness of the subspace of structured shapes (i.e., a principal component analysis showed that not only are such shapes rare, but they also all lie in the same, small region). These two properties may have important consequences from an evolutionary viewpoint, since (1) architectural morphogenesis is robust, and (2) the type of shapes that can be produced is highly constrained [20,21]. Finally, we found *a posteriori* that structured shapes can be built only with special algorithms, characterized by an "emergent" coordination [20,21]: stimulating configurations corresponding to different building stages must not overlap, so as to avoid the deorganization of the building activity. This feature creates implicit handshakes and interlocks at every stage, i.e. constraints upon which structures can develop. This coordination is essential, since parallelism — that has a clear adaptive advantage by allowing the colony to build at several locations at the same time — introduces the possibility of conflicts. Our approach therefore indicates how the nest itself can provide the constraints that canalize a stigmergic building behavior into producing specific shapes.

Our approach has been described at length in [1,20,21]. The aim of this paper is to present a few architectures obtained with cubic bricks on a 3D cubic lattice and some shapes obtained on a 3D triangular lattice with hexagonal bricks (more precisely, the lattice is triangular and the bricks are hexagonal in the x-y planes). Finally, we report the first results of a genetic algorithm (GA) applied to the exploration of the space of algorithms.

Fig. 1 Examples of architectures (cubic)

2. ARCHITECTURES

We now present a few examples of architectures grown by artificial agents moving randomly on a lattice, and performing very simple asynchronous actions with purely local information. Two types of bricks can be deposited. No brick can be removed once it has been deposited.

2.1 Cubic lattice and cubic bricks

In figure 1, all architectures, except *g* and *h,* have been obtained with coordinated algorithms. The difference between (*g*, *h*) and all other architectures is striking. One given coordinated algorithm always converges towards architectures that possess similar features. Any uncoordinated algorithm seems to diverge: the same algorithm leads to different global architectures in different simulations. This tendency to diverge comes from the fact that stimulating configurations are not organized in time and space and many of them overlap, so that the architecture grows in space without any coherence. As an illustration of this fact, architectures *d* and *e* result from two successive simulations using the same coordinated algorithm, and architectures *g* and *h* result from the same non-coordinated algorithm. We see that there is a high degree of structural similarity between *e* and *d*, in sharp contrast to *g* and *h*. Moreover, even in shapes built with coordinated algorithms, there may some degree of variation, which is higher in cases where the number of different choices within one given building state is high. For example, we see that *d* and *e* are similar but not perfectly identical: this is because there are several possible options at many different steps in the building process, or, in other words, the process is not fully constrained. Architectures *k*, *l*, *m*, and *n* provide examples of fully constrained architectures: two successive runs produce exactly the same shapes. It would certainly be useful to quantify the degree of similarity of architectures produced by the same algorithm in different runs.

As emphasized in [20,21], some of the structured architectures of figure 1 are reminiscent of natural wasps nests. Among these nests the presence of plateaus is observed in *b*, *d*, *e*, *f*, *i*, and the different levels of the nest are linked together with either a straight axis (*b*, *i*) or with a set of pedicels (*d, e, f*). Some others posses an external envelope: nests *k* and *n* are shown with a portion of the front enveloppe cut away so as to allow for a visualization of the nest's interior, and correspond to nests found in the genera *Chartergus*. Figures *c, g, h, j, l, m,* are examples of architectures not found in nature.

2.2 Triangular lattice and hexagonal bricks

One limitation, among others, of the model (if it is to be considered as a model of building in wasps) is that the simulated swarms 'live' in an abstract space using cubic bricks to build architectures, while natural wasp nests are built with hexagonal paper cells: symmetry properties are not equivalent. To overcome this limitation, we have run simulations with bricks that are hexagonal in the xy-planes (horizontal planes are triangular lattices). Figure 2 presents a few of architectures obtained with this particular topology. We can see that these shapes look more "biological" than shapes generated with cubic bricks.

Fig. 2 Examples of architectures (hexagonal)

3. GENETIC ALGORITHM

3.1 Introduction

The space of possible architectures grown by artificial agents on a lattice, and capable of depositing two types of bricks according to the local state of the environment, is huge and long to explore, even if one restricts one's attention to (at least partially) spatially isotropic rules. To overcome this difficulty, our exploration is based on a genetic algorithm (GA). Using a GA here seems appropriate as the mapping function from algorithms to architectures appears to be smooth. Because it is in the present case hard to define fitness functions on *architectural phenotypes* (it is difficult to define formally the adaptive value or the biological plausibility of the shape of a nest: what is an "interesting" architecture?), our fitness function is evaluated according to a heuristic criterion on behavioral phenotypes. An alternative choice would consist of using an interactive GA, whereby a human observer would evaluate the fitness of a pattern according to her taste or goal, and algorithms which generate the most ‘interesting’ architectures would be selected — the term ‘interesting’ may then refer to biological resemblance or to any other criterion, be it subjective or objective: as biologists we prefer biologically plausible architectures and as engineers we may be looking for ‘useful’ or simply appealing patterns. We chose a more automatic selection procedure because an interactive GA generates too many purely random shapes for the observer not to quickly become saturated.

Our fitness criterion is based on a simple observation: in algorithms that generate coherent architectures, many micro-rules are used, whereas in algorithms that generate structureless shapes, only one rule or a few rules are actually used in the course of the simulation. A given algorithm is defined by its rule table, which assigns an action to be taken to each configuration of bricks. If one is interested in, say, algorithms that contain 40 rules, some of these 40 rules may never be used, because the configurations of matter they correspond to never occur. Only algorithms many rules of which are actually applied in the course of the simulation can produce coherent architectures, because such architectures require a subtle and complex sequence of correlated deposits. Obviously, this condition is necessary but not sufficient, but a population of algorithms containing a large proportion of algorithms that are fit according to this criterion is certainly a much better point to start from.

3.2 Results

Because of high computational requirements, the results have been obtained after a few hundred generations (1 month of CPU on a DEC-α workstation) — only about 10^6 shapes have been evaluated (and analyzed afterwards by an observer). Algorithms, that is, rule tables of fixed length (40), are encoded into bit strings, and the behavioral fitness is computed by counting the number of different rules that have actually been used in a run. A given simulation stops either when a predefined proportion of space is filled (say 10% of the 20×20×20 grid; most algorithms generate random space filling quite rapidly), or after a predefined number of steps (say 5.10^5 steps, where one step corresponds to one agent taking one action). Simulations have to be long enough because it takes time to produce coherent architectures, owing to the numerous constraints they induce. The choice of rules in the initial population, and when new random rules are injected into subsequent generations, must be slightly biased to allow construction to start: (1) stimulating configurations containing a large number of bricks must be avoided because they may never be encountered in the course of one simulation (we implement this bias by using a rapidly decaying probability distribution for the number of bricks in a stimulating configuration), and (2) simulations start with one and only one brick in space, so that there must one rule whose associated configuration contains only one brick. Moreover, some rules that systematically induce a rapid — and random — filling of space can and must also be avoided. Finally, we use a random one-point crossover and a fixed mutation rate of 0.05 (more precisely, 5% of all rules are mutated at a single site). Each generation has a population of 500 algorithms.

Fig. 3 Fitness

Fig. 4 Example of architecture obtained with the GA

Despite these precautions, only a small number of coherent architectures are generated. Figure 3 presents the evolution of the maximum and average fitness (n° of applied rules divided by 2), and figure 4 presents a shape that has been found with the GA, the fitness of which is 8, which means that 16 micro-rules have actually been applied during the simulation. The saturation of fitness shows that coherent associations of rules may be destroyed by recombinations and/or mutations. In particular, the application of some rules requires the prior application of other rules. But the proportion of interesting shapes among the highest ranking algorithms is definitetely much larger than in random populations. This result suggests that (1) algorithms producing interesting shapes are very rare, and (2) the fitness function may not be the best one. Furthermore, despite the apparent smoothness of the mapping from algorithms to architectures (which can be an artefact of the procedure used to design algorithms), the fitness function itself may not be smooth, so that convergence to good solutions is unlikely.

4. CONCLUSION

A fair coverage of rule space requires many more simulations. One of the problems was to define a semi-automated procedure to undertake a systematic exploration. Our use of a GA combined with a heuristic criterion is one step in the right direction. On the biological side, this exploration is aimed at characterizing formally and logically the set of algorithms that generate coherent architectures: what property (or properties), if any, do these algorithms have in common? Our prior characterization, in terms of coordination, is somewhat biased, although plausible, because we used a very specific procedure to produce the algorithms [20,21]. This procedure was simply convenient and may not allow all "interesting" algorithms to be found. Knowing whether coordination is absolutely necessary to build coherent, complex architectures, or is simply a side effect of our original procedure may have important implications on our understanding how social insects collectively produce functional nests. Another important information is whether modularity, that we observe in all the coherent architectures we obtained, is a natural consequence of self-assembling processes, or if it is due to exogeneous factors (such as an intrinsic periodicity in building behavior, or, over a longer timescale, the colony's reproductive cycle), or is a combination of endogeneous and exogeneous factors.

In order to increase the efficiency of the exploration, several factors may be taken into account in the design of the GA or in the definition of a fitness function:

(1) Additional biases may be added to generated rules. For example, diagonal brick deposits should not be allowed, because they often lead to quick random space filling and are not necessary for interesting architectures. By diagonal brick deposit, we mean a deposit in which the deposited brick has only an adjacent edge (as opposed to an adjacent face) with already present bricks. Therefore, only stimulating configurations allowing the deposit of bricks which have at least one face in common with other bricks should be used.

(2) We have mentioned on several occasions that it takes more time to generate a complex architecture because of the constraints that are generated in the course of construction. In fact, each constraint acts as a bottleneck preventing any progress from being made until the constraint is satisfied. This result is interesting as it relates to the notion of logical depth, which is defined as the time is takes for a good program to generate a given pattern. Obviously, here, interesting architectures are logically deep. Time might therefore be taken into account in the fitness function, in combination with the number of micro-rules which are actually used.

(3) In contrast with algorithms that produce structureless shapes, those that produce interesting shapes are convergent: two successive simulations lead to similar shapes. The degree of overlap between shapes obtained with the same algorithm in different runs can therefore act as a good clue. But this notion of overlap has to be defined clearly, and even interesting algorithms may not be perfectly convergent. The simplest solution is to used a mere spatial overlap, so that fully constrained architectures would be favored. Note however that simple columns (where bricks are deposited under each other) are often encountered in the simulations, are not particularly interesting, but exhibit a perfect overlap from one simulation to the next (because they are fully constrained). This criterion should therefore be used in conjunction with others. Finally, it might be useful to define overlap not only in space, but in spacetime, as the order in which building actions are taken is important.

Our model, and the exploratory tool we developed, can be useful to understand phenomena outside the realm of social insects:

(1) It can apply to the formation of sponges, or to cell sorting.

(2) Let us stress that the model is a generalization of diffusion-limited aggregation (DLA) and other growth models. In DLA, a randomly moving brick sticks to a cluster as soon as the cluster is met: DLA is a specific case of the model, where more complicated rules can be implemented.

(3) The model can help study the question of whether "robust" morphogenesis is generic. That is, are interesting shapes stable with respect to modifications of the algorithms that produce them? In that context, it may also be useful to extend the model to include "noisy" behavior (that is, for example, incorrect perception of configurations, etc.).

Until now, we have mentioned only real insects, that is real biological systems. But the problem of coherent distributed building has counterparts in computer science and engineering: the same principles can be used to explain the behavior of existing biological systems, or/and to design new artificial devices based on them. Our simple computer simulation can be seen as a simplified model of collective building in social insects, or alternatively as an experiment carried out on artificial agents whose building abilities we wanted to test. We clearly see that this computer model can have interesting consequences for biologists and for engineers: biologists hope to gain some understanding of the algorithmic processes involved in, and required by building behavior, through simulating minimal algorithms followed by simple agents; such simulations can provide engineers with valuable insight when they need to design programs or machines exhibiting collective problem-solving abilities, such as self-assembling machines or robots (areas of current potential applications already include microelectronics, optics, microelectromechanical systems and displays [2]). In both cases, biologists and engineers are faced with the same 'inverse problem': given a desired shape, or more generally a desired state, find the simplest or a fairly simple behavioral algorithm that can generate it. Within this context, we believe it worthwhile to develop correspondence tables between simple algorithms and the shapes that they can produce. Bowden et al. [2] recently reported the design of two-dimensional arrays made out of mesoscale objects (from 1 mm to 1 cm) by self-assembly: there, the shapes of the assembling objects and the wettability of their surfaces determine the structure of the arrays (the objects are floating at the interface between perfluorodecalin and water, and interact by lateral capillary forces). Our approach allows to deal theoretically with even more complicated cases, where the objects have more local (but still not a lot of) "intelligence": one can hope to be able to implement these more intelligent units by small-scale simple microprocessors, that would connect to one another if neighborhood conditions are satisfied.

ACKNOWLEDGEMENTS

This work was supported in part by a grant from the GIS (Groupe d'Intérêt Scientifique) Sciences de la Cognition. E. B. is supported by the Interval Research postdoctoral fellowship at the Santa Fe Institute.

REFERENCES

[1] Bonabeau, E., Theraulaz, G., Arpin, E. and Sardet, E. (1994). The building behavior of lattice swarms. In Artificial Life IV (Brooks, R. and Maes, P., eds.), pp. 307-312, MIT Press.

[2] Bowden, N., Terfort, A., Carbeck, J. and Whitesides, G. M. (1997). Self-assembly of mesoscale objects into ordered two-dimensional arrays. Science 276, 233-235.

[3] Deneubourg, J.-L. (1977). Application de l'ordre par fluctuations à la description de certaines étapes de la construction du nid chez les termites. *Ins. Soc.* 24, 117-130.

[4] Deneubourg J.L. and Goss S. (1989). Collective patterns and decision making. *Ethol.Ecol.and Evolution* 1, 295-311.

[5] Deneubourg, J.-L., Theraulaz, G. and Beckers, R. (1992). Swarm-made architectures. In: *Toward a Practice of Autonomous Systems, Proceedings of The First European Conference on Artificial Life* (Varela, F.J. and Bourgine, P., eds) pp. 123-133. Cambridge, MA: The MIT Press/Bradford Books.

[6] Downing, H. A. and Jeanne, R. L. (1988). Nest construction by the paperwasp Polistes: A test of stigmergy theory. *Anim. Behav.* 36, 1729-1739.

[7] Downing, H. A. and Jeanne, R. L. (1990). The regulation of complex building behavior in the paperwasp *Polistes Fuscatus* . *Anim. Behav.* 39, 105-124.

[8] Franks, N. R., A. Wilby, V.W. Silverman and C. Tofts. (1992). Self-organizing nest construction in ants: sophisticated building by blind buldozing. *Anim. Behav.* 44, 357-375.

[9] Gallais-Hamonno, F.G. and R. Chauvin, 1972. Simulations sur ordinateur de la construction du dôme et du ramassage des brindilles chez une fourmi *(Formica Polyctena)*", *C. R. Acad. Sc. Paris* 275 D, 1275-1278.

[10] Grassé, P.-P. (1959). La reconstruction du nid et les coordinations inter-individuelles chez *Bellicositermes Natalensis et Cubitermes sp*. La théorie de la stigmergie : essai d'interprétation du comportement des termites constructeurs", *Ins. Soc.* 6, 41-81.

[11] Grassé, P.-P. (1984). Réparation, reconstruction et remaniements internes du nid. Coordination des tâches individuelles et comportement stigmergique. Le déterminisme du comportement constructeur. In: *Termitologia, Tome* II – *Fondation des Sociétés – Construction* – (Grassé, P.P. ed.), pp. 490-577. Paris: Masson.

[12] Hansell, M. H. (1984). *Animal architecture and building behavior*. Longman: London.

[13] Jeanne, R. L. (1975). The adaptativeness of Social wasp nest architecture. *The Quaterly Review of Biology* 50, 267-287.

[14] Karsai, I. and Penzes, Z. (1993). Comb building in social wasps: self-organization and stigmergic script. *J. Theor. Biol.* 161, 505-525.

[15] Karsai, I. and Theraulaz, G. 1994. Nest building in a social wasp : postures and constraints, (submitted).

[16] Kugler, P.N. and Turvey, M.T. (1987). *Information, natural law, and the self-assembly of rhytmic movements*. Hillsdale, NJ: LEA.

[17] Kugler, P.N., Shaw, R.E.,Vincente, K.J. and Kinsella-Shaw, J. (1990). Inquiry into intentional systems I: Issues in ecological physics, *Psychol. Res.* 52.

[18] Skarka, V., Deneubourg, J.-L. and Belic, M.R. (1990). Mathematical Model of Building Behavior of *Apis mellifera*. *J. Theor. Biol.* 147, 1-16.

[19] Theraulaz G., Goss S., Gervet J. and Deneubourg J.-L. (1991). Task differentiation in *Polistes* wasp colonies: a model for self-organizing groups of robots. In: From Animals to Animats, Proc. of the 1st Intnl. Conf. on Simulation of Adaptive Behavior (Meyer, J.A. and Wilson, S.W., eds), 346-355, MIT Press.

[20] Theraulaz, G. and Bonabeau, E. (1995). Coordination in distributed building. *Science* 269, 686-688.

[21] Theraulaz, G. and Bonabeau, E. (1995). Modeling the collective building of complex architectures in social insects with lattice swarms. J. theor. Biol. (in press).

[22] Wenzel, J. W. (1991). Evolution of nest architecture. In: *Social Biology of Wasps* (Ross, K. G. and Matthews, R.W., eds) pp. 480-521. Ithaca: Cornell University Press.

[23] Wilson, E.O. (1971). *The Insect Societies*. Cambridge, MA: Harvard University Press.