

AN EVOLUTIONARY DESIGN PROCESS

Adaptive-iterative explorations in computational embryogenesis

Vignesh KAUSHIK and Patrick JANSSEN
National University of Singapore, Singapore
vigneshkaushik@gmail.com, patrick@janssen.name

Abstract. Computational embryogenies are a special kind of genotype to phenotype mapping process widely used in explorative evolutionary systems as they provide the mechanism for generating more complex solutions. This paper focuses on how designers explore embryogenies for specific design scenarios through an adaptive-iterative process. The process is demonstrated for a complex project to generate a prototypical urban farm in Singapore. It is shown that by employing an adaptive-iterative process, the embryogeny can be made progressively more complex and less abstract, thereby allowing the exploration to be guided by the designer.

Keywords. Computational embryogeny; evolutionary; multi-criteria optimization; encoding; decoding.

1. Introduction

Evolutionary design algorithms evolve populations of designs by iteratively applying a set of procedures to design variants in the population (Frazer, 1995). With the evolutionary design approach, the parameters for a design variant are referred to as the genotype (with each individual parameter being a gene), the model of the design variant is referred to as the phenotype, and the evaluations of the design variant are referred to as performance scores. Three key procedures need to be defined: a developmental procedure that generates a phenotype from a genotype, one or more evaluation procedures that calculate performance scores, and a feedback procedure that performs genetic reproduction based on the performance scores.

For designs with limited variability, the developmental procedure can use direct parametric modelling. However, in cases where greater variability is required, the developmental procedure may require a more complex transformation process.

These complex types of developmental procedures are commonly referred to as ‘computational embryogenies’ (Bentley and Kumar, 1999).

Embryogenies are a special kind of genotype to phenotype mapping process widely used in explorative evolutionary systems as they provide the mechanism for generating more complex solutions. This paper focuses on how designers explore embryogenies for specific design scenarios through an adaptive-iterative process. Previous research by Bentley and Kumar (1999) on computational embryogenies has typically focused on their behaviour and efficiency of search and their scalability for evolving different morphologies. More recently, both Dillenburger et al. (2009) and Janssen and Kaushik (2013) have proposed evolutionary design approaches for the automatic arrangement of building volumes on a given site. Although generative and evolutionary design is an on-going research topic, little attention has been paid to the process of how designers might create such embryological procedures for specific design scenarios.

As a demonstration of this process of exploring an appropriate embryogeny, this paper presents a case-study of creating an embryogeny for generating complex spatial configurations for a thesis project at the National University of Singapore.

In the case study example presented in this paper, the process of constructing an embryogeny involved four versions of adaptive-iterative exploration, each described in more detail in Section 4.

2. Design Scenario

The design scenario focuses on the need for Singapore to become more self-sufficient in terms of food production, since it imports over 90% of its food requirement (Osman, 2011). Hence, as a long term strategy to ensure food resilience for Singapore’s growing population, a prototypical urban farm typology catering to a population of 10,000 people is proposed. The idea would be to have a decentralised network of such urban farms across various parts of Singapore.

State-of-the-art farming methods would allow essential food items such as vegetables and fruits, fish and chicken to be grown vertically and simultaneously within the same building. The urban farmers who grow the food were to live within the same complex as well. Also, in order to produce a part of the energy to power the building, it was decided to grow algae (bio-fuel) on photo-bio-reactor pipes to be fixed to the parts of the façade receiving the most solar radiation. The market, seed storage, waste recycling centre and many other allied spatial systems were to be located within the complex.

Each of these spatial functions has individual daylight requirements and adjacency rules that must be satisfied for its optimum functioning. For example, the

vegetable growth chamber must have adequate daylight, whereas the chicken growth chamber requires controlled daylight. The latter should preferably be far away from the farmers' housing but closer to the markets. The fish growth chambers need to be at lower floors for structural reasons but as close to the vegetable growth chambers as possible. Such complexity of the spatial and functional inter-relationships and constraints means that a complex embryogenic type of developmental procedure is required.

2.1. TOOLS

The developmental and the evaluation procedures were defined using a Visual Dataflow Modelling (VDM) approach using Houdini, an advanced procedural CAD application (Janssen and Chen, 2011a). Modelling in a VDM system consists of creating dataflow networks using nodes and links. Each node can be thought of as a function performing an action and a link is used to connect the output of one function to the input of another function. VDM systems are increasingly being used as an important tool in performance-based design approaches (Shea et al., 2005; Coenders, 2007; Lagios et al., 2010; Toth et al., 2011; Janssen and Chen, 2011b; Janssen and Kaushik, 2012).

The evolutionary algorithm was executed using Dexen, a distributed execution environment. The feedback procedure, generated automatically by Dexen, will rank groups of phenotypes using a standard Pareto ranking method, and will then create new genotypes using standard crossover and mutation operators (Janssen et al., 2011).

3. Exploring Embryogenies

An embryogeny is a process of growth that defines how a phenotype is generated from a genotype. For example, in nature, embryogenies indicate how an animal should be grown. Hence the genotype may be regarded as a set of 'growing instructions', or a recipe that defines how a phenotype will be developed.

Current computational embryogenies can be classified into three different types: external, explicit and implicit (Bentley and Kumar, 1999). External embryogenies can be imagined to be a piece of computer code that performs a one to one mapping from genes to parameters and the process is not generally evolved. An explicit embryogeny is where every step of the growth process is explicitly specified as instructions in the data structure. Typically, the genotype and the embryogeny are combined and are allowed to evolve simultaneously. The third type, the implicit embryogeny, does not explicitly specify each step of the growth process, but is implicitly coded by a set of rules or instructions. By evolving a set

of simple rules which can then be iteratively applied to each element of the growing solution, it is believed that many large scale problems can be tackled.

For the purpose of this design scenario, versions of varying complexities of external embryogenies were used for evolutionary exploration. These versions became progressively more complex and less abstract. One of the reasons for using external embryogenies is that the user retains more control over the final evolved form. Moreover, one can constantly improve the quality of evolved designs by making careful modifications to the embryogeny. However, one must ensure that this complex mapping process will always produce a legal phenotype and overcome a common issue with such complex developmental procedures, referred to as the variability problem (Janssen, 2004).

3.1. VERSION 1

For the first exploration, the design problem discussed above was split into a set of cubes that were allowed to float on an abstract site of 250m by 250m. Each cube represented a fragmented part of the various functions in the urban farm and was spatially positioned within a 3d grid, as shown in Figure 1.

The embryogeny used a direct representation for defining the positions of each of the cubes within the 3d grid. The genotype consisted of a set of real-valued

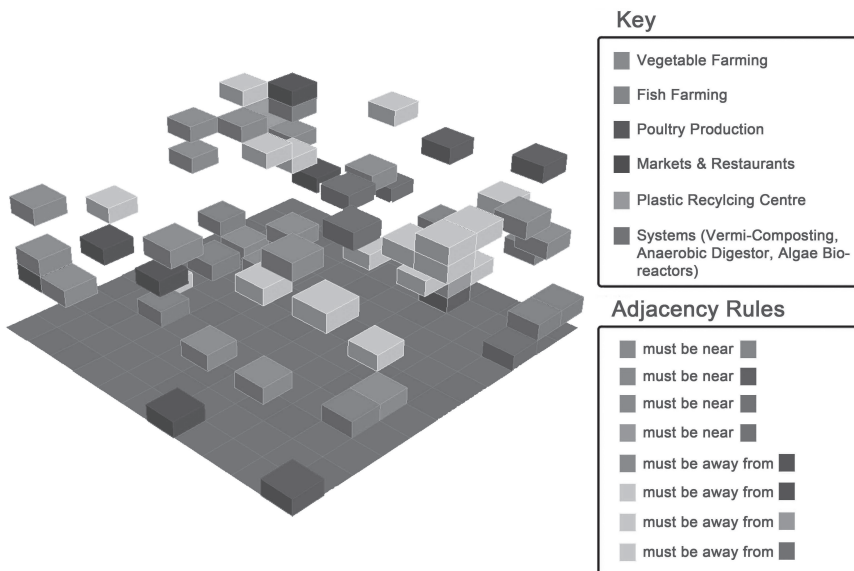


Figure 1. Floating cubes – version 1 of embryogeny exploration.

genes in the range $\{0,1\}$. For each cube, the position was defined by three genes, which were mapped to a 3d coordinate position in the grid. No constraints whatsoever were set and hence cubes were allowed to float in space and multiple cubes were allowed to occupy the same position in the grid. Each phenotype was then allocated an overall fitness score by evaluating certain simple adjacency rules for each cube. The performance criteria for the evolutionary algorithm were to maximise the number of cubes satisfying the adjacency rules. An example of one of the design variants is shown in Figure 1.

The aim of this experiment was to acquire an overall understanding of the behaviour of the cubes under the influence of the various adjacency rules. It was expected that the unconstrained freedom of position of each cube relative to all others would allow a variety of promising spatial patterns to be identified. However, although the exploration produced many phenotypes that satisfied most of the adjacency rules, it was very difficult to evaluate them visually and understand their behaviour due to their high variability. The main reason identified for the chaotic variability was the lack of constraints. Hence, in the next stage, additional constraints were introduced.

3.2. VERSION 2

In version 2, two constraints were introduced: cubes were not allowed to float, and multiple cubes were not allowed to occupy the same position in the grid. The same genotype representation was used, consisting of three real-valued genes in the range $\{0,1\}$ for each cube. Two genes were mapped to a 2d coordinate position in the grid and the third gene was used to define the stacking order of the cubes. All the cubes with the same 2d coordinate were sorted according to the stacking gene, and were then stacked in that order, starting from the ground up. An example of one of the design variants is shown in Figure 2.

As expected, certain patterns and groupings emerged among the various functions. However, the variability among the solutions remained chaotic. Two key deficiencies were identified: the vertical circulation of the cubes was not clear due to the lack of cores, and the floor areas required for each function was not constrained.

3.3. VERSION 3

From the solutions generated from the previous versions, it was decided to group certain functions into a tower and podium typology. The tower consisted of four functional types; farmer's housing, vegetable farms, chicken farms and fish farms. All other allied functions were grouped together to form the podium. It was also decided to opt for a smaller site of 150m by 180m, with real surroundings that would have a bearing on the evolution of a podium and tower building. For the

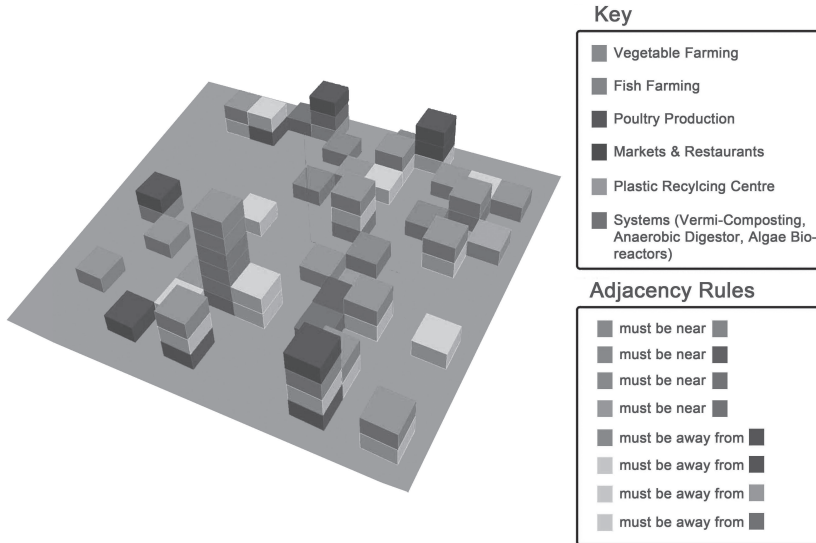


Figure 2. Stacking cubes – version 2 of embryogeny exploration.

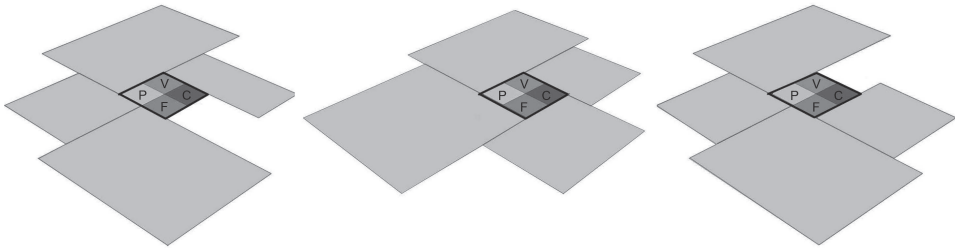
purpose of simplifying the search space, the focus of version 3 was only to evolve the tower, assuming a fixed podium block as a base of the tower.

The tower's core consisted of four independent sub-cores (one for each function in the tower) with each catering to its respective part of the floor plan. These sub-cores were structurally integrated but functionally independent. This ensured that at any floor level, functions could be arranged in a flexible manner.

The developmental procedure used a combinatorial parametric modelling technique for generating the floor plans. Each floor consisted of four rectangles of varying sizes, with different functions assigned to each rectangle. The genotype consisted of a total of 240 genes, 12 genes per floor and three genes per rectangle. One gene was used to select the shape of the rectangle from a set of predefined possibilities; one gene was used to indicate the orientation of the chosen rectangle around its sub-core, and one gene was used to select the function of the rectangle. Three variants of the floor plan configuration are shown in Figure 3.

The fitness scores were based on a number of evaluation criteria, such as sufficient daylight for the food growing chambers and the farmer's housing, scenic views and privacy for the housing, and percentage of façade that could be utilised for growing algae with photo-bio-reactor pipes.

It was also decided to add a constraint that would limit the floor areas assigned to each of the four functions. The required floor areas are known in advance, and a technique was therefore required for defining this as a constraint within the evolutionary



*Figure 3. Variants of floor plan configuration.
P-farmer's housing, V-vegetable farm, C-chicken farm, F-fish farm*

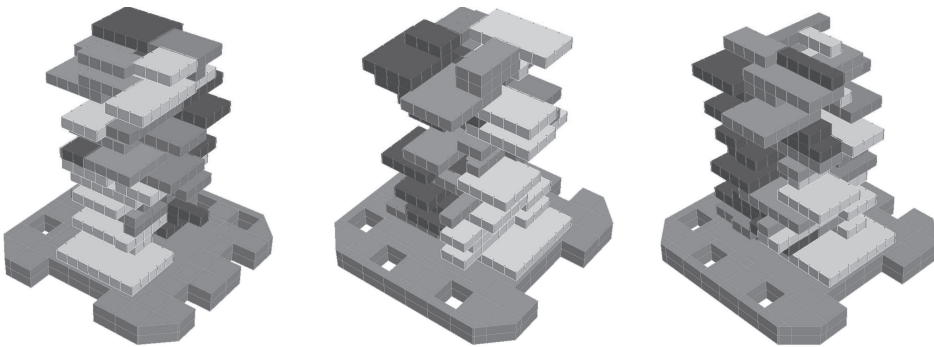


Figure 4. Phenotypes from version 3 of embryogeny exploration.

system. Researchers have identified four main approaches to handling constraints in evolutionary algorithms: 1) penalty functions 2) repair functions, 3) specialised reproduction operators, and 4) specialised genotype to phenotype decoder functions (Eiben and Smith, 2008). For this version, a penalty function was added that reduced the fitness of those solutions where the area assigned to each function was not desirable.

The solutions from the evolutionary exploration of this version indicated that certain aspects of a tower design such as the structural grid had to be refined. In addition, the penalty function resulted in many variants with very low scores, which degraded the ability of the evolutionary algorithm to evolve high performance designs. Figure 4 shows three variants from the population of evolved designs.

3.4. VERSION 4

After analysing the results from the previous version, it was decided that the size and shape of the floor plan should be defined in multiples of a specific structural

grid. After careful analysis, a structural grid of 7.2m x 7.2m was adopted as it was found to be well suited for all the four functions in the tower.

It was also noted that the use of a penalty function for constraining the floor areas of the different functions was not successful. In this version, this constraint was embedded within the developmental procedure, using a rule based decoder function that combined direct and indirect mapping of genotypes to phenotypes. This approach falls under the fourth category of constraint handling techniques as described by Eiben and Smith (2008). In this case, the genotype to phenotype mapping process was structured as a sequential chain of decisions (for more details see Janssen and Kaushik, 2013).

Each floor consisted of four rectangles of varying sizes, each overlapping with one of the sub-cores. The dimensions of each rectangle were defined as a multiple of 7.2m. As with the previous version, the rectangles were assigned different functions. The genotype consisted of a total of 180 real-valued genes in the range $\{0,1\}$, 9 genes per floor. The first 8 genes were used to specify the dimensions of the rectangles. The genes were mapped to integer values in the range $\{0,4\}$ (resulting in a maximum of four grids of 7.2m each from the edge of the core). This mapping procedure is shown in Figure 5.

The ninth gene was used to select how the different functions were assigned to the four rectangles. In total, there were 21 different ways of assigning functions to the rectangles. Six of the 21 configuration are shown in Figure 6. However, in order to constrain the floor area assigned to each function, a filtering process was first performed. Those configurations that would result in excess floor area being assigned to any of the four functions were filtered out as being invalid. The ninth gene was then used to select one of the remaining valid configurations by mapping it to an integer value in the range $\{0,n\}$, where n was the total number of valid configurations. This ensured that all design variants were assigned approximately the required floor areas for the four functions.

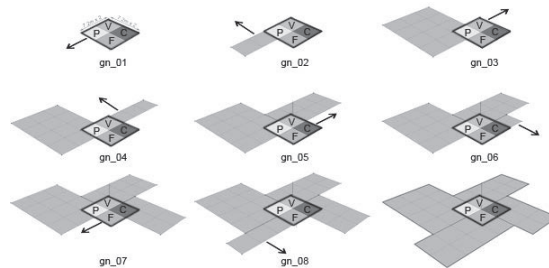


Figure 5. Developmental procedure of version 4 of embryogeny exploration.
P-farmer's housing, V-vegetable farm, C-chicken farm, F-fish farm.

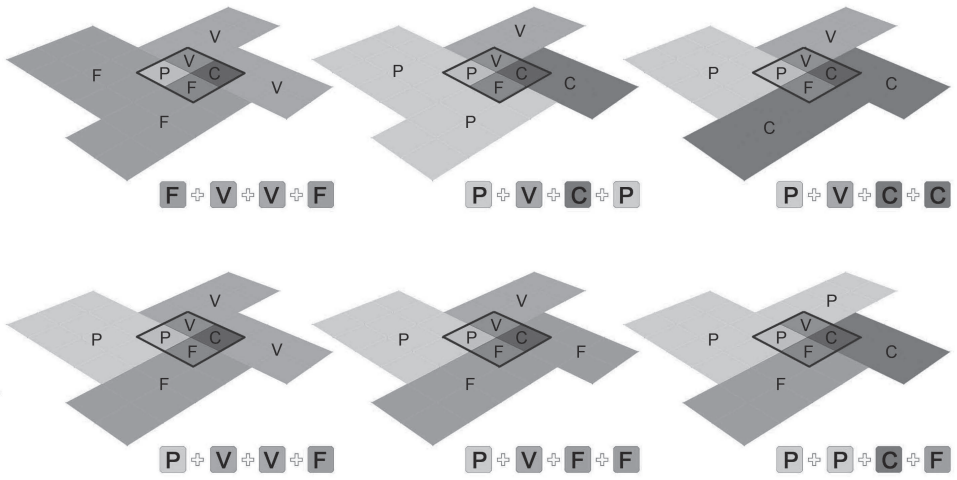


Figure 6. Few options of assigning the functional configurations. P-farmer's housing, V-vegetable farm, C-chicken farm, F-fish farm

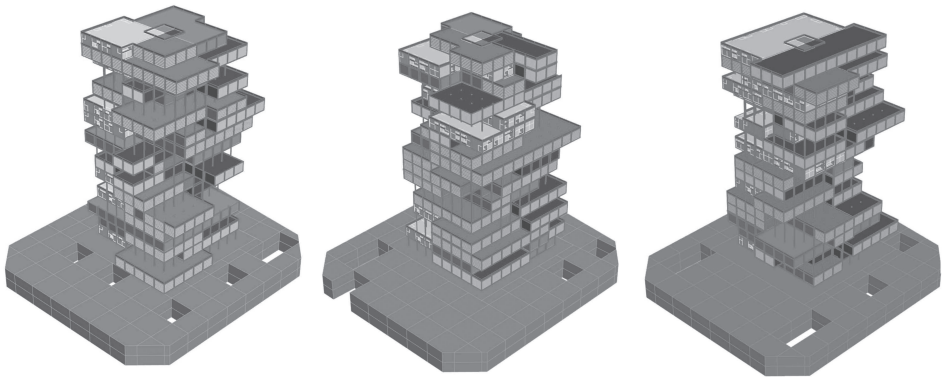


Figure 7. Phenotypes from version 4 of embryogeny exploration.

In this version, the control of variability was improved significantly and as a result the evolutionary exploration produced solutions with better performance. Figure 7 shows three examples of evolved designs.

4. Conclusion

This paper focussed on the process of creating embryogenies for a complex design problem through a sequential process of adaptive-iterative exploration. At each

version, the solutions that were evolved were not interpreted as optimal answers, but as diagnoses of potential problems and as suggestions for further architectural explorations (Caldas et al., 2001). Also as Dawkins (1988) points out, certain embryogenies are better than others at producing certain morphologies and hence it is essential for designers to explore such trade-offs through a trial and error process. By employing an adaptive-iterative process, the embryogeny can be made progressively more complex and less abstract, thereby allowing the exploration to be guided by the designer.

References

- Bentley, P. and Kumar, S.: 1999, Three ways to grow designs: a comparison of evolved embryogenies for a design problem, *Proceedings of the Genetic and Evolutionary Computation Conference*, Morgan Kaufmann, 35–43.
- Caldas, L. and Rocha, J.: 2001, A generative design system applied to Siza's school of architecture at Oporto, in J. S. Gero, S. Chase and M. Rosenman (eds.), *Sixth Conference on Computer-Aided Architectural Design Research In Asia (CAADRIA'01)*, Sydney, Australia, 253–264.
- Coenders, J.: 2007, Interfacing between parametric associative and structural software, in Y. Xie and I. Patnaikuni (eds.), *Proceedings of the 4th International Conference on Structural and Construction Engineering*, Taylor & Francis, London, 63–68.
- Dillenburger, B., Braach, M. and Hovestadt, L.: 2009, Building design as individual compromise between qualities and costs: a general approach for automated building generation under permanent cost and quality control, *Proceedings of the 3rd International Conference on Computer Aided Architectural Design Futures*, 458–471.
- Eiben, A. E. and Smith, J. E.: 2008, *Introduction To Evolutionary Computing*, Springer.
- Frazer, J.: 1995, *An Evolutionary Architecture*, AA Publications, London.
- Janssen, P. H. T.: 2004, A design method and computational architecture for generating and evolving building designs, PhD Dissertation, The Hong Kong Polytechnic University.
- Janssen, P. and Chen, K. W.: 2011a, Visual dataflow modelling: a comparison of three systems, *Proceedings of the 4th International Conference on Computer Aided Architectural Design Futures*, Liege, Belgium, 801–816.
- Janssen, P. H. T. and Chen, K. W.: 2011b, Iterative virtual prototyping: performance based design exploration, *Proceedings of the 29th eCAADe Conference*, Ljubljana, Slovenia, 253–260.
- Janssen, P., Chen, K. W. and Basol, C.: 2011, Evolutionary developmental design for non-programmers, *Proceedings of the 29th eCAADe Conference*, Ljubljana, Slovenia, 245–252.
- Janssen, P. H. T. and Kaushik, V.: 2013, Decision chain encoding: evolutionary design optimization with complex constraints, *Proceedings of the 2nd EvoMUSART Conference*, Vienna, Austria, 157–167.
- Lagios, K., Niemasz, J. and Reinhart, C. F.: 2010, Animated building performance simulation (ABPS) – Linking rhinoceros/grasshopper with radiance/daysim, *Proceedings of SimBuild '10*, New York.
- Osman, M. M. B.: 2011, “Speech by Dr Mohamad Maliki bin Osman, Senior Parliamentary Secretary for National Development and Defence at the opening ceremony of the International Conference on Asian Food Security”. Available from: Ministry of National Development <<http://app.mnd.gov.sg/Newsroom/Speeches.aspx/>> (accessed 15 February 2011).
- Shea, K., Aish, R. and Gourtovaia, M.: 2005, Towards integrated performance-driven generative design tools, *Automation in Construction*, **14**(2), 253–264.
- Toth, B., Salim, F., Burry, J., Frazer, J., Drogemuller, R. and Burry, M.: 2011, Energy-oriented design tools for collaboration in the cloud, *International Journal of Architectural Computing*, **9**(4), 339–359.