

Interactive Artificial Life Based Systems, Augmenting Design Generation and Evaluation by Embedding Expert Opinion

A Human Machine dialogue for form finding.

Subhajit Das¹

¹Georgia Institute of Technology

¹das@gatech.edu

Evolution of natural life and subsequently selection of life forms is an interesting topic that has been explored multiple times. This area of research and its application has high relevance in evolutionary design and automated design generation. Taking inspiration from Charles Darwin's theory, all biological species were formed by the process of evolution based on natural selection of the fittest (Darwin, n.d.) this paper explains exploratory research showcasing semi-automatic design generation. This is realized by an interactive artificial selection tool, where the designer or the end user makes key decisions steering the propagation and breeding of future design artifacts. This paper, describes two prototypes and their use cases, highlighting interaction based optimal design selection. One of the prototypes explains a 2d organic shape creator using a metaball shape approach, while the other discusses a spatial layout generation technique for conceptual design.

Keywords: *design generation, implicit surfaces, artificial life, decision making, artificial selection, spatial layout generation*

INTRODUCTION

Design teams work meticulously to find design options to satisfy client requirements and to propose optimal solutions given project constraints. However, with the traditional design process, finding an optimal design is limiting, as only a handful number of design options can be explored. This leaves many optimal and virtuous design options unexplored. In addition, this prohibits the designer to explore new design directions. On the other hand, with the cur-

rent prowess of modern machines, there are seminal work which shows substantial design intelligence can be added to the machine, to find an optimal design from the vast design search space (Adriaenssens, et al., 2014) (Anon., 2017) (Kai-Uwe & Ekkehard, 2001)

However, these algorithms are strictly objective, requiring the designer to frame the constraints and goals of the problem as an objective fitness function i.e. find an optimal design which has low solar gain

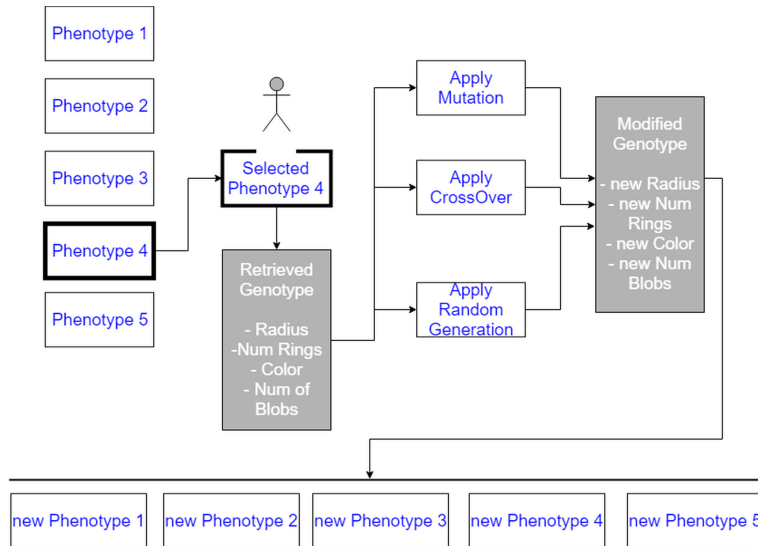


Figure 1
Diagram showing the interactive artificial selection process, showing how genotypes are modified based on selections made by the user.

and high rainfall catchment area. However, in practice the design process and the choices that are made are often subjective (Thurston, 1990) (Raharjo, et al., 2008) both by the designer and by the client. People select designs because they like them for some factors, which often cannot be summarized or captured in an objective function. Another limitation with current design optimization techniques are that, it does not involve the designer to steer the optimization process in direction which they deem have potential. Genetic algorithm techniques is time consuming to compute, during which the designer has to wait till the system responds with a list of ranked optimal design solutions. Many of the solutions or options evaluated are often useless for the designer as the user is not able to intervene to guide the machine in a more promising direction. Design goals often change dynamically, as designers explore alternatives early on, in the design process. Not being able to steer the direction of search is a serious limitation of the current state of the art optimization processes. This makes genetic algorithms limiting to find the true goal of the designer. However, there are some examples of

interactive genetic algorithm techniques which have explored capturing subjective nuances of the user, as shown in the work of (Cho & Lee, 2002). However, they face issues such as, difficulty to handle complex models including lack of enough genetic operators and user fatigue and uncertainty of decision making. Often these optimization models lack an adequate explanation for the type of options they generate etc.

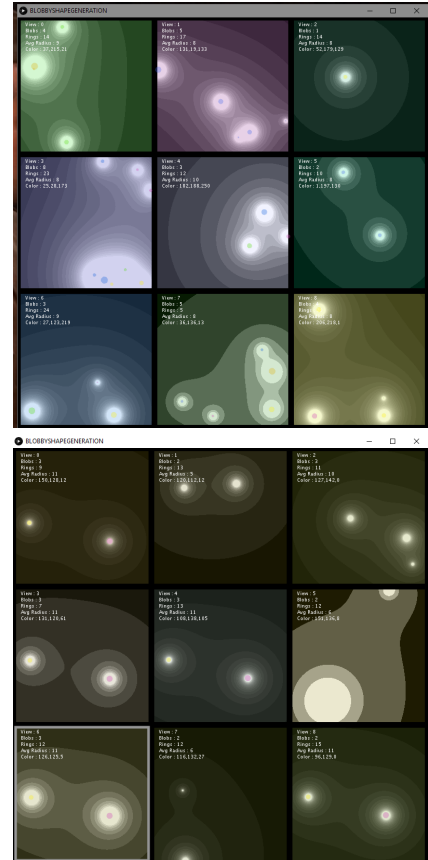
To this end, I took inspiration from Charles Darwin's, theory of natural selection (Darwin, n.d.) and Richard Dawkins simulations of artificial life (Dawkins, 1996) as an alternative solution. According to Darwin's theory of evolution, natural selection is the process which describes that organisms capable of adapting to the dynamically changing conditions of their environment get to survive while the others are naturally eliminated in the process of evolution. Supporting the theory of natural selection, Dawkins conducted numerous Artificial Life simulations, including The Blind Watchmaker program. From his experience from the simulations, he added that the nature evolved by a not so careful design process, but by random gene mutations and non-random survival

Figure 2
Shows the meta ball shape option generation based on system recommendation and user selection. Every rectangular block is a selectable option for the user.

Figure 3
Updated view of the design options, when the user makes a selection to the previous iteration.

(Dawkins, 1988). His Blind Watchmaker program is a sophisticated computer model of artificial selection supporting his argument. The simulation modeled a “bimorph,” represented by straight lines, defined by its, length, position and angle. The formation of such creatures was simple rule-driven, very similar to a genome. The rules could add new lines, change their position, and angle, leading to a discrete number of creature variants, selectable from the screen by a user. Whichever variant the user would pick, becomes the basis for next generation of form mutation by changing the selected genome in various ways. Dawkins argued that the selection made by the user could very well be the random selection by nature leading to the formation of new variants of organisms giving rise to random complexity.

Extending this interactive workflow for design search, I propose an “Interactive Artificial Selection” based design option generation process, which allows the designer to be an active participant in the optimal design search process. Simulating the natural selection process in evolutionary biology, a design artifact is comparable to a phenotype (the biomorph from the Blind Watchmaker program, (Dawkins, 1996)), which has certain characteristics or in computational term: “parameters.” These parameters are encoded as genotypes or genome. The system represents and builds phenotypes using genotypes. The process to translate a genotype to form a phenotype is called embryology, while the capability to copy or morph genotypes by mutation is called genetics (Kumar & Bentley, 2003) (Dawkins, 1988). The user explores the shown phenotype options and picks an option she likes. Next, the system reconciles the genotype of the selected phenotype and formulates new phenotypes from it. The process to form new phenotype includes crossover, mutation and with randomly regeneration of genotypes. Refer Figure 1. Our idea is to develop a computational framework, a tool generating design options of artifacts based on an embedded grammar or language or rulesets.



In the next section, the paper describes literature review and relevant works in the direction of design optimization. Then it describes two use cases of the proposed technique, highlighting how it can benefit designers by bringing them closer to the search and retrieval of optimal designs from the vast design search space. I summarize the paper with a discussion and conclusion section. This paper has the following contributions : 1. Extension of a well-defined technique from evolutionary computing, highlighting interactive steering of machine intelligence to search optimal or desirable design solutions 2. A sys-

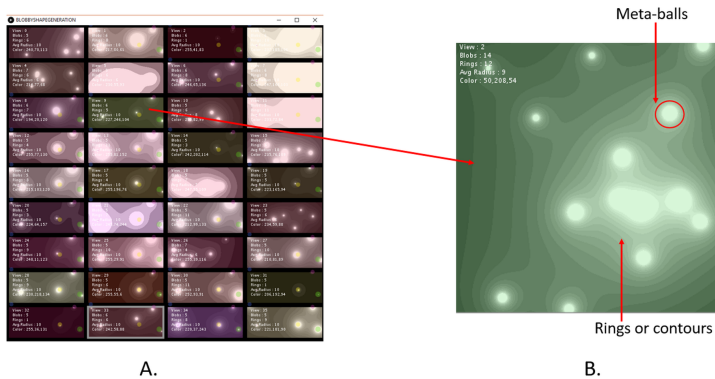


Figure 4
 (A) Showing 36 designs off metaballs visually composed in different ways. (B) Explains the component of a design option, defined by a metaball, contours and its color.

tem prototype, showcasing design option generation and selection based on the extended technique described. 3. Two use cases; one with a blobby shape generator and another to demonstrate spatial packing in concept design stage.

RELATED WORKS

Genetic Algorithm

Cho et al. implemented an interactive genetic algorithm to build an image retrieval system using human preference and emotion. Their system extracted image features from a large database, in spite of the user not being able to define what the image should be clearly. Their search and retrieval technique supported implicit queries based on emotion reflected in the image. They used interaction as a means to evaluate the search results, as it was impossible to define an explicit fitness function for implicit (subjective) queries. However, they lacked enough capabilities to encode correct expression of images and likewise needed to test additional genetic operators (Cho & Lee, 2002). Xiayan et al. studied user evaluation fatigue and uncertainties in complex optimization scenarios. They tested an evaluation technique of user fatigue in interactive optimization methods for a personalized search for books. Supporting a user based on an evaluation of optimization results

instead of explicit fitness function (often impossible, due to the subjective nature of personalized search space), the authors designed a technique to reduce human fatigue and uncertainty in the interaction pipeline. Their technique analyzed browsing time and user preference using a Gaussian model, which performed better than previous other techniques (Xiayan, et al., 2017). My work gets inspiration from the above, in providing a dialogue between the user and the machine enabling a workflow capable of capturing user's subjective choices and dynamic design goals iteratively.

Design Option Generation

Eiben et al. have looked into evolutionary computation's role in hardware development and autonomous machines adaptive to their environments. They compare evolutionary computation with natural evolution, highlighting its benefits concerning other methods. Also, they emphasise generative representation of phenotypes allowing reuse of code, enabling scaling of complexity in varied areas (Eiben and Smith 2015). Karl Sims showed his work in using genetic algorithms to generate morphologies of creatures and their neural systems by employing varied fitness functions. He used nodes and connections representing directed graphs to structure the morphology and the neural circuitry. He fur-

ther mated the graphs to form a variety of design morphology by crossover technique. He also compared complexity with control and discussed the role of aesthetic selection as a possible means to control the production of these creatures as opposed to allowing the automatic evolution of the forms. (Sims, 1994). Prior to this, Karl Sims modeled genotypes as symbolic expressions in an attempt to surpass the limitations of fixed-length genotypes with predefined expression rules. He further reinstated unlike genetic algorithms, his technique of creating varied forms, creatures and shapes were not dependent on an explicit analytic function measuring any fitness. He moved away from traditional genetic algorithm method, as it is difficult to automatically measure the aesthetic visual qualities of simulated objects or images etc. Hence he relied on a human user to provide that judgement, demonstrating prototypes using combinations of automatic and interactive selection (Sims, 1991). Work of Mark Bedau studies the overlap of cognitive science and artificial life in the field of molecular self-organization, evolutionary robotics and evolutionary complexity and language. One of the important aspect studied was evolvability which depicts the capacity of evolution to create new phenotypic variation and the systems ability to search the variations. (Bedau 2003). I am using some of these techniques in the design space for architects and graphic designers and this paper describes two prototypes proving that with modern machines, we can encode an interactive artificial selection based optimization loop where the user is the key to steer the search direction of the underlying model.

SYSTEM DESCRIPTION: SOFT BLOBBY SHAPE CREATION

In order to simulate a natural process of design generation, there needs to be some rules and guidelines laid out. These rules can propagate creation of genotypes, which gets translated to a phenotype, the process called as embryology (Dawkins, 1988). Once the process of design generation is in place, it can give rise to the emergence and organic complexity. For

the test case, I have selected implicit shape based blobby objects or metaballs (Blinn, 1982) to create organic forms in 2d space. The system initiates by automatically generating some design artifacts for the user to explore. As aforementioned each such design artifact is called a phenotype, and the genes that create them are called genotype (Sims, 1991). Genotypes are the encoded information set to build the phenotype. It can be procedural parameters, symbolic expressions or binary strings. A set of procedural rulesets or preset operators transforms an input phenotype to a new phenotype. After every design generation, the user selects one or few of the output designs (or phenotypes, Refer Figure 2). At the next iteration, the system consumes the selected phenotype as input and develops new phenotypes based on the selected one. This is facilitated by random mutation and crossover between the choices selected by the user (the expert) which is explained further in the paper.

Phenotype and Genotype: Implicit Shape driven Blobby Objects

Implicit shapes are generated by equations of form $F(x,y,z) = c$, where c is a constant. I used the following equation to generate the meta ball shapes:

$$F(x, y, z) = \frac{r \cdot r}{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}$$

where, r is the radius of the balls, x_0, y_0, z_0 is the center of the metaball. Based on the metaball creation algorithm [1], I used a random threshold value to ascertain an isosurface, which encloses the space inside or outside the boundary formed from the equation. Genotypes are represented as a sequence of parameters, which can be used to produce a future phenotype (Sims, 1991). Genotypes have two important properties. One is that they can be copied from an existing phenotype to the other, and the other property is being able to change their values, to create new genotypes from existing ones. These properties are also termed as Genetics by Dawkins in his artificial life simulations. The success or failure of any phenotype from one generation to the next is dependent on the qualities embedded in its genotype. In this use case,

part of genotypes descriptions are a number of blobs "nb", radius "r", distance "d", color-value "cl", number of contour lines or rings "nr" and a sequence of operators. For example: (radius, colorValue, distance) (add)(split)(extend). Operators are described in the next section.

Operators

Once a 2d composition of metaball based soft shapes with contour lines are rendered as shown in Figure 4, I deployed additional operators, which enable the user to modify any selected design option. As mentioned previously, the operators are part of the genotype description. Any of the methods added below becomes part of the genotype. The operators are discussed below:1. Add: This adds a new metaball to the existing base phenotype.2. Subtract: This removes the last metaball from the existing base phenotype.3. Split: This adds a metaball in between two metaballs at a random location on the line connecting the two metaballs.4. Extend: This adds a new metaball on an extended line between any other nearby connected metaball.5. Angular Add/Void: Adds or removes a metaball at an input angle about one of the input coordinate axes.

Mutation

Mutation is a process, which ensures variation in the future production of design artifacts (Whitley, 1994). Mutation is a random procedure. Whenever the system starts mutating, it changes the genotype by a random factor. The updated genotype transforms the phenotype into a newly generated design artifact, which varies from the original by some characteristics (Refer Figure 5).

Cross Over

As the user makes a selection of designs, the system randomly picks some of the previously selected design choices as parent phenotypes and mixes their genotypes in various ways (Whitley, 1994) to form new genotypes as shown in Figure 5. This is similar to mating in organisms to form a possible hybrid candidate. The proportion in which different genotypes are combined is randomly chosen for each generation, ensuring variation in new design generation. For example, as shown in Figure 3 design with 5 small red toned metaballs is cross-bred with 2 balls large, green-toned metaballs to come up with new design variants.

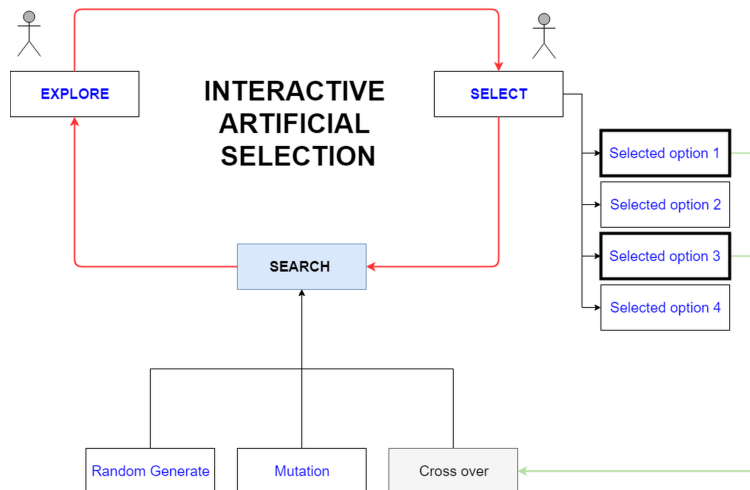


Figure 5 Shows the Interactive Artificial Selection cycle. Clarifies parts handled by a human, while the search is handled by the machine. Explains the cross-over techniques based on past set of selected designs by the user.

Fitness Function

The described approach is very similar to the genetic algorithm-based search processes (Grenfenstette 1987). It differs in one main aspect that the next generation of improved design artifacts are not based on a fitness function or objective function but are based on selective choices made by end users. I call this process, “interactive artificial selection.” However, I also tested interactive artificial selection with the objective function based selection, meaning that the users can define their objective function early on. As the system iterates through generations, it shows possible design alternatives and waits for the user to make a selection. Next, it breeds the next generation. Simultaneously, it automatically highlights design artifacts which are deemed optimal based on the performance of the objective function. This enables the user to compare between option selected subjectively and option recommended objectively in the system.

Designers Use

A system like above can assist a designer to find a design pattern for graphic design or other creative purposes, such as interior tiling, facade pattern formation, etc. Such a design goal, often would not have objective metrics to evaluate each generation. An interactive tool like above provides a workflow, where the designer makes choices from possible options and updates design goals incrementally. At the same time, the tool performs like an idea generator of possible solutions, which haven't been thought of before. Another similar use case could be site planning and landscaping option generator, where the goals are little more objective. The position of the metaballs could be the built space, while the contours could be landscape profiles or change of levels. An example can be seen in Figure 4. Iterative selection can quickly provide many design alternatives in the direction that the designer thinks is interesting.

SYSTEM DESCRIPTION: SPATIAL LAYOUT DIVISION

Interactive Artificial Selection based technique can be tested on form generation and early space packing ideas. In this prototype the designed artifact or the phenotype is a built form, with packed departments or zones: a collection of spaces which can be grouped functionally in a built space). Refer Figure 6. An example of a zone and a space in a hotel tower is that the hotel bed tower is a zone, while each hotel room is space. Similarly, Food and Beverage is a zone while, kitchen, back of house services, maintenance room, mechanical room is space. The system would generate a phenotype of a 2D form (shown as a closed outline), subdivided by polygons representing zones. It also communicates the attributes of the phenotype, for example, how many required zones are fitted in the spatial layout, the total area covered, My prior work, Space Plan Generator (SPG) provides a solution to this problem of automated space plan layout generation (Das, et al., 2016). However, unlike SPG, this prototype is iterative and interactive in design choice selection following similar principles as mentioned above. Also, the technique allows to search similar options to the one that the user liked and thus can help find similar looking space plan layout variations.

Phenotype and Genotype : A Form Generation

The phenotype here is the built form represented as a 2D closed polyline. The user inputs a required area and list of departments or zones each with an area requirement. The system breaks down the form in $1 \dots n$ rectangles, which are called “parts”. Each part is given a proportioned size $ps_1 : 0.3, ps_2 : 0.7$. The system stores a list of length to width ratios i.e., $rt : \left\{ \begin{matrix} sqr : \frac{1}{1}, rectA : \frac{1}{2}, rectB : \square, elong : \frac{3}{1} \end{matrix} \right\}$. Given the number of parts, the system assigns an area to each part and randomly picks a ratio for it. Each part has four points: p_1, p_2, p_3, p_4 . A point “pi” has four sides namely $A, B, C,$ and D . At

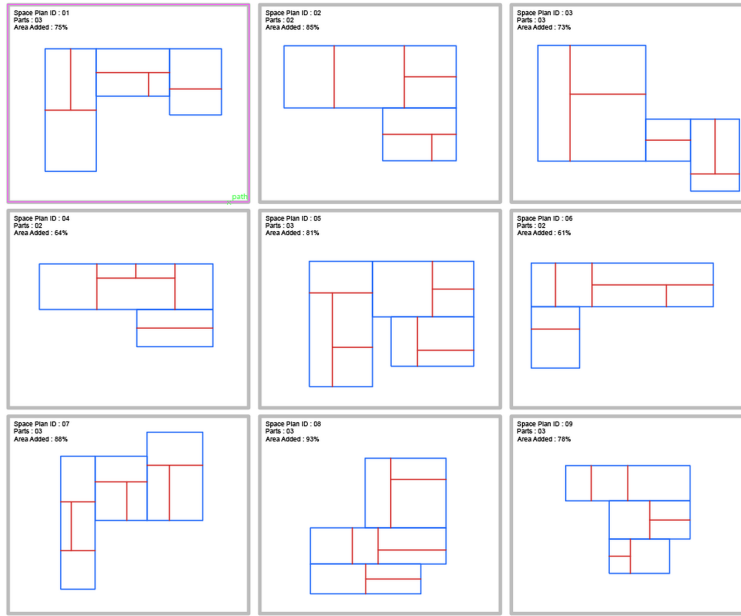


Figure 6
Image showing the spatial layout options. The blue outline is the parts for the form, while the inner red line represents composition of zones inside to satisfy the user requirement.

any point, the parts are placed from left to right and defined by an encoded genotype. The left part is placed with an orientation, i.e. 0: landscape and 1: portrait. Then the next part is attached at one of the point “pi” along one of its sides *A*, *B*, *C* or *D* with a set orientation. This is repeated till all parts are added. The net form is defined by the total of all the parts. The genotype is encoded with the code, part-Name|pointName|sideName|orientation. An example genotype code is 00|p2|C|L02|p3|A|P01|p1|A|P. Refer Figure 7.

Phenotype and Genotype: A Zone Creation

At the next iteration, the system fits zones in the form. For each part, the system picks zones of the comparable area and fits zones using a strategy used in work (Das, et al., 2016). It would pick the smaller side of the part rectangle and add a rectangle of the required area representing the zone added. The system would iterate till all parts are filled with zones or required zones to be added already added. This technique of-

ten might under or over satisfy a zone area, but at this point, it's acceptable as the intent of the technique is to provide approximate conceptual spatial packing options as shown in Figure 8.

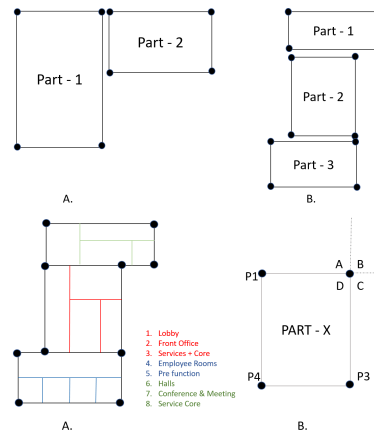


Figure 7
Diagram (A) and (B) shows how different parts can integrate to make a form for the spatial layout.

Figure 8
(A) Shows the spatial organization of parts and zones within each part. (B) Shows for each part there is four points and any such point has four sides.

Operators

Given the phenotype, the system allows the user to specify certain actions using a preset list of operators. Actuating any of them updates the genotype of the design enabling future replication and mutation of the gene. The following operators are added:

- Add: This adds another part to the phenotype. The part attaches to the right of the last added part, to any randomly picked point.
- Remove: Removes the last part added to the spatial layout. System reconciles to fit the zones in the removed part to the other existing parts.
- Rotate: The user can change the orientation of any part or can change the orientation of the total spatial layout.
- Elongate: Can elongate the longer side of any of the part from the spatial layout. The system extends the size of the zones fit in it.

Designers Use

A system like above is clearly useful in the early concept design phase for Architects and Space Planners. It helps them ideate solutions to complex layout problems, without the need to draft or sketch solutions. Reducing time to explore design alternatives not only helps them explore more options in a short time, but also allows them to invest project resources efficiently. Steering the production of layouts in the direction they deem fit, is unique to this prototype as opposed to some of the past such tools as aforementioned.

Discussion

The technique described allows the designer to control the direction of search to the system (Refer pseudo code provided). However, the prototypes described have few limitations. One limitation is that there is no memory or stored checkpoint. At any point, if the user likes some options but, wants to steer in a different direction to explore, there is no way to revert to the same option. One possible solution could be if the system can save snapshots of the saved states including a collection of all phenotypes and their genotypes, the user can retrieve previously liked design artifacts. Another limitation is that when the change in the genotype is not significant enough,

there is not enough variation in the design options. The user explores almost same design alternatives, leading to a dead end in the otherwise large search space. A possible solution can be adding an interactive slider to the interface, allowing users to control the randomness in new genotype creation.

```
Algorithm describing phenotype
↳ generation process.
//function to make phenotypes
function makePhenoOptions(numOptions){
for(var i =0;i<numOptions;i++){
    var selectedPheno =
        ↳ getUserSelectedPheno();
    var selectedGeno =
        ↳ getUserSelectedGeno(
            ↳ selectedPheno);
    var updatedGeno = updateGeno(
        ↳ selectedGeno);
    var newPheno = makePheno(updatedGeno
        ↳ );
    phenoDisplayList ,push(newPheno);
}
}

//function to update genotype from a
↳ given genotype
function updateGeno(givenGeno){
var newGeno;
var randomNum = getRanomNumberBetween
↳ (1,0);
if(randomNum >=0.7 && randomNUM< 1){
    newGeno = mutateGeno(givenGeno);
}else if(randomNum >=0.4 && randomNum
↳ < 1){
    newGeno = crossOverGeno(givenGeno ,
        ↳ randomGeno);
}else{
    newGeno = randomGeno();
}
return newGeno;
}
```

Another possible future work would be coupling interactive genetic algorithm and interactive artificial selection. The system thus formed can keep showing optimal design alternative based on an objective fitness function, while the user iteratively selects design

options based on above techniques. This would facilitate comparison of optimal design options based on objective and subjective criteria. Possibly the final selected option could be a mix of the best from both ends. As the simulation runs, on any generation, users can mix the two set of options by a given proportion, asking the system to generate those options, which the user likes based on their selection at each iteration but also picking the ones which performed well in the objective criteria. Furthermore, the process allows the system to capture and integrate user's subjective choices or bias with the objective criteria of design generation. The systems currently do not account for symmetry in the visual arrangement. However, it will be interesting to encode symmetry in the genotype of the design options.

Conclusion

Interactive artificial selection based generation of design output shall augment designers work with computers to generate, explore and evaluate designs that they like and think is a better-suited solution for the context. Till now, computers and humans have been working independently in areas where they are good. But the research presented starts a dialogue between the human and the machine enabling the efficient use of compute resources. Unlike many other similar systems comprising of endless searches to find fitting solutions including genetic algorithm based systems, the solution shown will allow designers be part of the optimization process and allowing capture subjective criteria of the user. As a test case, I prototyped a modeling fluid metaball based soft shapes to aid design delivery of organic forms, but this can be extended further in other ways to the device and implement design generation grammar. Another example I tested is juxtaposing programmatic components as polygons representing spaces or rooms.

REFERENCES

Sigrid Adriaenssens, PB 1999, *Shell Structures for Architecture: Form Finding and Optimization*, Routledge
 Bletzinger, KU and Ramm, E 2001, 'Structural optimization and form finding of light weight structures',

Computers & Structures, 79(22), pp. 2053-2062
 Blinn, JF 1982, 'A Generalization of Algebraic Surface Drawing', *ACM Trans. Graph.*, 1(3), pp. 235-256
 Cho, SB and Lee, JY 2002, 'A human-oriented image retrieval system using interactive genetic algorithm', *IEEE Trans. Systems, Man, and Cybernetics, Part A*, 32, pp. 452-458
 Darwin, C 1985, *The Origin of Species*, New American Library
 Das, S, Day, C, Hauck, A, Haymaker, J and Davis, D 2016 'Space Plan Generator: Rapid Generation & Evaluation of Floor Plan Design Options to Inform Decision Making', *Proceedings of Acadia 2016*, Ann Arbor, USA
 Dawkins, R 1986, *The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe without Design*, Turtleback Books
 Dawkins, R 1988, 'The Evolution of Evolvability', in Langton, C (eds) 1988, *Artificial Life, SFI Studies in the Sciences of Complexity*, Addison-Wesley Publishing Company, pp. 201-220
 Guo, Z and Li, B 2017, 'Evolutionary approach for spatial architecture layout design enhanced by an agent-based topology finding system', *Frontiers of Architectural Research*, 6(1), pp. 53-62
 Sanjeev Kumar, PJB 2000, *On Growth, Form and Computers*, Elsevier
 Raharjo, H, Brombacher, A and Xie, M 2008, 'Dealing with subjectivity in early product design phase: A systematic approach to exploit Quality Function Deployment potentials', *Computers & Industrial Engineering*, 55, pp. 253-278
 Sims, K 1991 'Artificial evolution for computer graphics', *SIGGRAPH*
 Sun, X, Chen, Y, Bao, L and Xu, R 2017, 'Interactive genetic algorithm with implicit uncertainty evaluation for application in personalized search', *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1, pp. 1-8
 Thurston, L 1990 'Subjective design evaluation with multiple attributes', *Design Theory and Methodology*, pp. 355-361
 Whitley, D 1994 'A genetic algorithm tutorial', *Statistics and Computing*, pp. 65-85
 [1] https://www.siggraph.org/education/materials/HyperGraph/modeling/metaballs/metaballs_mward.html