Novelty Search for Shape Descriptors

Simon Hickinbotham
Dept. of Electronic Engineering
University of York
York, UK
simon.hickinbotham@york.ac.uk

Róisín McConnell
School of Mechanical and Aerospace Eng.
Queen's University Belfast
Belfast, Northern Ireland
r.mcconnell@qub.ac.uk

Imelda Friel
School of Mechanical and Aerospace Eng.
Queen's University Belfast
Belfast, Northern Ireland
i.friel@qub.ac.uk

Mark Price
School of Mechanical and Aerospace Eng.
Queen's University Belfast
Belfast, Northern Ireland
m.price@qub.ac.uk

Wei Zhang
School of Mechanical and Aerospace Eng.
Queen's University Belfast
Belfast, Northern Ireland
w.zhang@qub.ac.uk

Andy Tyrrell
Dept. of Electronic Engineering
University of York
York, UK
andy.tyrrell@york.ac.uk

***Abstract*—In nature, the exploration of a design space is achieved by evolution. However, using artificial evolution to evolve physical shapes has been challenging because both the mapping from the genotype to phenotype and the means of measuring the resulting shape to estimate fitness are not straightforward. This contribution brings together recent advances in generative design with novelty search, an evolutionary method that replaces the fitness function with reward based purely on novelty. Bringing these techniques together yields a new methodology for exploring the power of shape descriptors. Novelty search with and without an archive is used to explore the range of shapes that are reachable from a hand-designed genotype, and compared with random walk mutations. Results indicate that the novelty search technique without an archive evolves a wider range of shapes than when an archive is used, but both are better than random walk.**

***Index Terms*—generative evolution, novelty search, genotype-phenotype mapping**

I. Introduction

Innovation in product design is crucial, not just to ensure market competitiveness but to meet increasingly stringent engineering and manufacturing requirements. A typical product development process begins with a concept which in effect is a shape/form of a final product. Subsequent processes will reduce the design space further to converge on the final design that meets the product requirements and thus satisfying the goal.

A key drawback of this approach is that innovation is stifled, because designers are heavily influenced by preceding products and new designs are often mere perturbations of existing ones. To avoid predicted solutions, a new design system and philosophy are required, such as generative design [1], [2], which takes an additive approach that highly encourages divergent exploration and exploitation with no predefined solution [3].

Generative design requires an algorithm to drive the search for good designs, since brute-force evaluation of the entire search space is usually intractable. Evolutionary algorithms (EAs) are a good fit for such tasks as they are easily configurable and generally able to find good solutions in complex search spaces. However, the application of EAs to generative design remains challenging. A suitably flexible genomic representation must be found, along with a genotype-phenotype (GP) mapping. In addition, a set of descriptors of the resulting shape must be found that are capable of measuring the fitness of the resulting phenotype. For product design, a means of incorporating feedback from the environment into the fitness measure is necessary to ensure that the phenotype is fit for purpose.

Each of the components of a generative design platform is critical to the overall performance of the system, but it is difficult to evaluate each of them simultaneously. One approach is to carry out a series of case studies on different aspects of this challenge. One such case study is presented here: the flexibility of the G-P mapping is explored using novelty search [4], which is known to work well where the G-P mapping is nontrivial. This allows the mapping and the search technique to be studied independently of various engineering constraints that can be introduced in subsequent work.

A simple grammar is used in this work, developed from Zhang et al [5], which has an interesting combination of grammar-like growth rules and position-based firing of these rules. There are many systems which use recursive firing of rules to generate complex forms, such as L-Systems and their variants [6]. However, in these systems it becomes difficult to specify particular features at particular locations, making it difficult to engineer useful objects which would need these features to perform some vital function. The representation used here combines the recursive feature generation properties of the more natural systems with a facility to add particular features at exact locations.

This approach aims to further enhance the generative design philosophy by creating a genotype as a design gene that will create a geometric shape from specific parameters in the gene. To date this work has successfully created a suitable phenotype that has met engineering requirements, but not as a direct response to them, rather it is limited by preset parameters in the gene that have controlled the emergence of

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

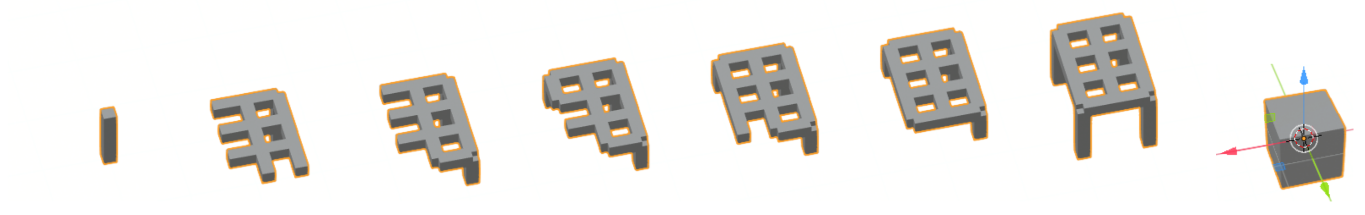

Fig. 1. Stages in the development of the phenotype as it develops from the genotype in table I using algorithm 1. Orientation in x (red), y (green) and z (blue) are indicated in the figure on the right.

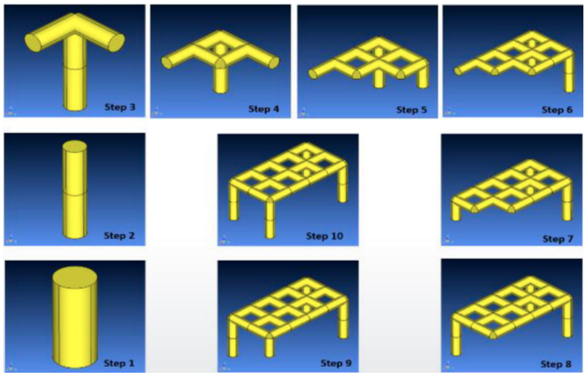

Fig. 2. Developmental stages of the original shape from [5]

this “table shape” phenotype. The challenge then becomes how to evolve these representations whilst preserving something of the essential characteristics that are required.

The key advantage of novelty search is that the emphasis of selection is not on fitness, but on descriptors which give a measure as to whether a similar or identical phenotype has been seen previously [7]. However, it appears that there is as yet little consensus about the best way to configure the search [8] even in established applications. The experiments presented here compare two representative configurations of this approach, one using an archive of phenotypes and one in which the search is based solely on the current generation. As a baseline, a run with the same population size and number of generations is used where there is no selection at all, to determine the phenotypes that could be achieved by random changes to the genotype.

The experiments presented below aim to give a methodology for the evaluation of the G-P mapping and EA parameters within the wider framework of generative design. A quantitative and qualitative interpretation of the approach is provided, with the goal of identifying and characterising the critical features of a new design paradigm.

II. Methods

In order to explore these ideas both a genomic representation of shape and the genotype mapping must be defined. These are both challenging tasks, particularly in the application to engineering structures. In addition, a series of metrics that are used to compare shapes is also needed. Without this, it is not possible to make any decisions regarding selection in an evolutionary algorithm. Even novelty search requires each member of the population to be placed in a space over which the search is carried out. Finally, an evolutionary algorithm is needed to bring each of these components together.

A. Genome

Each gene in the representation is a tuple of the following pieces of information. These values are coded directly, although there is scope for a more homogeneous genomic representation. The fields of each gene tuple are as follows:

- **Name:** the name of the gene - used for reference only
- **Type:** a text string that indicates how the gene is to be interpreted. This string can be parsed with regular expressions.
- **Value:** a text string giving a specifier to the gene type
- **Start & Stop:** the range of values on the X,Y or Z axis within which the gene is active. Where necessary, the axis is indicated by the last letter of the type string.
- **Dominance** is used where more than one gene could be expressed at a particular position - the gene with the higher dominance value takes precedent in this situation, and *only* this gene is expressed.

Note that in this representation the dominance value is used to separate genes into groups of three – one gene for each axis. This representation is easily extendable to incorporate environmental feedback in the growth process by adding extra fields where appropriate.

The genome that is used to initialise the experiments is shown in table I. This gene was designed by hand, to give a realistic representation of a table shape that might have utility in an engineering application and to give an identical baseline for each treatment. Other strategies such as random initialisation are feasible. The phenotype-shape that is described by these genes can be seen in the right hand image of figure 1.

There are eight groups of three genes. Genes 1 to 3 define the basic shape unit that is used throughout, since the work reported here concerns shapes made from a single shape primitive. The remaining blocks of genes express different portions of the final shape, as shown in figure 1. Genes 4 to 6 define the first leg at position [0,0,0]. Genes 7 to 9 form two sides and the centre of the horizontal portion of the shape. It has a branching nature of “L” shapes that is repeatedly expressed to form the table top. Genes 10 to 12 commence construction of the second corner at the far end of the Y axis and the far border, and genes 13 to 15 do the same for the

Name	Type	Value	Start	Stop	Dominance
gene01	“Cross Section”	“Square”	-40	40	1
gene02	“Length”	5	-40	40	1
gene03	“Diameter”	0.5	-40	40	1
gene04	“X_1X”	0	0	1	50
gene05	“Y_1Y”	0	0	1	50
gene06	“Z_1Z”	1	0	5	50
gene07	“T1_2X”	1	0	5	48
gene08	“T1_2Y”	1	0	10	48
gene09	“T1_2Z”	0	0	15	48
gene10	“T3_2X”	1	0	1	47
gene11	“T3_2Y”	0	15	16	47
gene12	“T3_2Z”	-1	6	10	47
gene13	“T4_2X”	0	10	11	46
gene14	“T4_2Y”	1	0	4	46
gene15	“T4_2Z”	-1	6	10	46
gene16	“T5_2X”	0	10	11	45
gene17	“T5_2Y”	1	4	14	45
gene18	“T5_2Z”	0	6	10	45
gene19	“L3_2X”	1	5	9	44
gene20	“L3_2Y”	0	11	15	44
gene21	“L3_2Z”	0	6	10	44
gene22	“L4_2X”	0	0	30	43
gene23	“L4_2Y”	0	0	15	43
gene24	“L4_2Z”	-1	1	12	43

TABLE I
GENOME FOR THE INITIAL SHAPE. GENES ARE GROUPED BY THEIR DOMINANCE VALUE.

far end of the X axis of the shape. Genes 16 to 18 and 19 to 21 specify the edges of the table along the far X and Y axes respectively. Finally, Genes 22 to 24 grow the lower parts of the three unfinished legs of the table shape. The resulting shape is similar in design and development to Zhang et al's original design [5], which is shown in figure 2 for comparison.

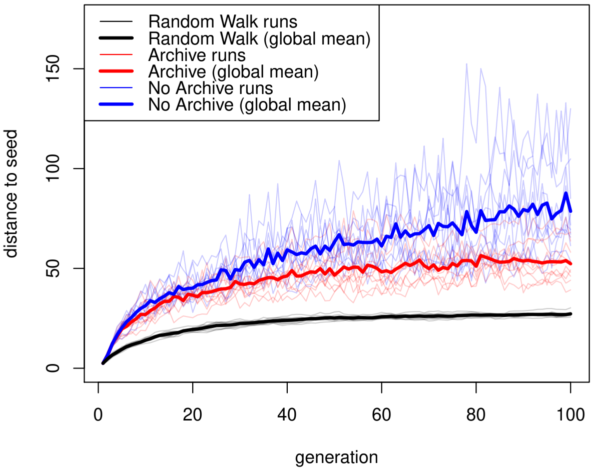

Fig. 3. Change in mean distance d of the population of phenotypes for the three different experimental approaches. Higher values of ‘distance to seed’ on the y-axis indicate better exploration of the shape space.

B. Genotype-phenotype mapping

The development algorithm is based on Zhang et al's work [5] but with some important simplifications. Firstly, no use is made of the “active” field, allowing all control of the firing of a gene to be governed by the dominance of the gene relative to the current position in processing the shape. Secondly, the x, y, z values and start positions are compounded into three genes rather than nine to control the direction of firing and the zone of dominance.

Algorithm 1 Genotype-phenotype mapping algorithm

```
procedure GPMAP($g, p, N$)
    $rp \leftarrow p$
    $n \leftarrow 1$
    while $p \neq \emptyset \wedge n \leq N$ do
        $nextp \leftarrow \emptyset$
        $n \leftarrow n + 1$
        for $pp \in p$ do                          ▷ for each position
            $s \leftarrow$ GETSHAPE($pp, g$)
            $d \leftarrow$ DOMATPOS($pp, g$)
            if $d \neq \emptyset$ then
                $V \leftarrow$ GETGROWTHVECTORS($d, g$)
                for $v \in V$ do
                    $nextp \leftarrow nextp \cup$ ADDPOSITION($s, v$)
                    $render(s, v)$
                end for
            end if
        end for
        $p \leftarrow nextp$
        $rp \leftarrow rp \cup nextp$
    end while
    return $rp$                ▷ $rp$ is the set of visited positions
end procedure
```

Algorithm 1 shows how a genotype is rendered into a phenotype. A genotype is processed by iteratively adding shape primitives to new coordinates as they are reached. The start, stop and dominance values determine which genes fire at each new position as it is reached. The following subroutines are needed: GETSHAPE scans the genome for the dominant shape description at the current position. Since there is only one shape primitive, this is trivial. DOMATPOS returns the dominant growth genes at this position. GETGROWTHVECTORS uses the genome and the dominance value to determine the directions of growth at the current position. ADDPOSITION identifies any new positions that may have been generated by growth. To prevent infinite loops of growth, each position is visited only once so a position is not added if it has already been processed.

To illustrate this process, figure 1 shows the seven stages of development of the shape genotype described in table I. Starting from x, y, z position $[0, 0, 0]$ one of the legs of the table is constructed first, as described in section II-A. The iteration limit N was set to 100 even though only nine iterations of the processing loop were needed to generate this shape. This approach was taken to impose a reasonable limit on computation time but to allow sufficient processing to generate rich phenotypes.

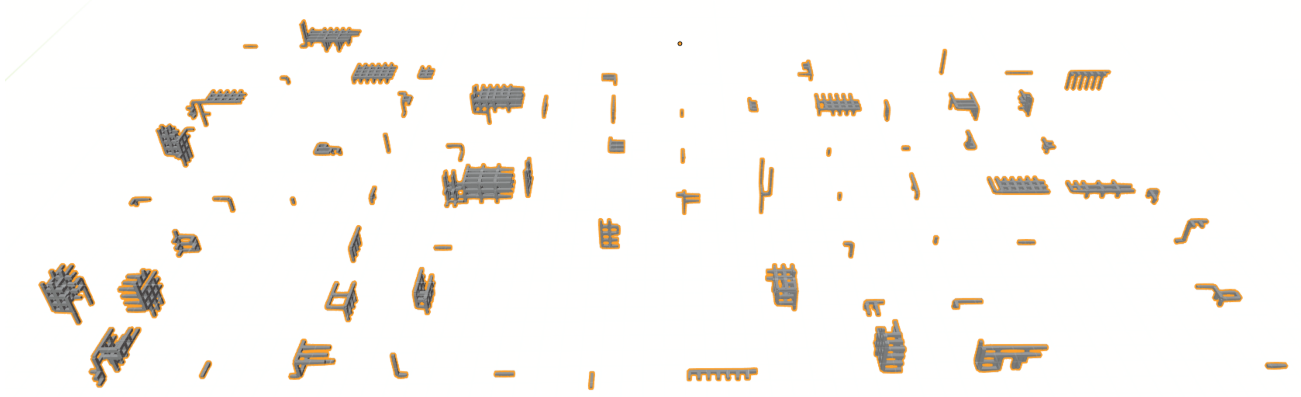

Fig. 4. State of the population of shapes after 100 generations using random walk

Algorithm 2 Novelty search algorithm

```
procedure NOVSEARCH($g, p$)
    $popsize \leftarrow 100$
    $elite \leftarrow 5$
    $ngen \leftarrow 100$
    $K \leftarrow 15$                    ▷ Generate the initial population
    for $pp = 1 \dots popsize$ do
        $gpop[pp] \leftarrow$ MUTATEGENE($g$)
        $ppop[pp] \leftarrow$ GPMAP($gpop[pp]$)
    end for
    for $gg = 1 \dots ngen$ do
        for $pp = 1 \dots popsize$ do
            $npop[pp] \leftarrow$ SHAPEMEASURE($ppop[pp]$)
        end for
        for $pp = 1 \dots popsize$ do
            $dpop[pp] \leftarrow$ MEANKDIST($npop[pp], K$)
        end for
        $rank \leftarrow$ ORDER($dpop$)
        for $ee = 1 \dots elite$ do                ▷ select elites
            $next[ee] \leftarrow gpop[rank[ee]]$
        end for
        for $pp = (elite + 1) \dots popsize$ do▷ Tournaments
            $ppair < -$SAMPLE($popsize, 2$))
            $next[pp] \leftarrow$ MUTATEGENE(WINNER($ppair$))
        end for
        $gpop \leftarrow next$
        for $pp = 1 \dots popsize$ do
            $ppop[pp] \leftarrow$ GPMAP($gpop[pp]$)
        end for
    end for
end procedure
```

C. Shape metrics

The G-P mapping is too unwieldy to act as the basis for a direct comparison of different phenotypes. Algorithm 1 returns a set of unique points that have been visited during processing of the genotype. These can be used to generate a set of metrics that allow numerical comparison between different shapes, which in turn can be used in evolutionary experiments. As a proof of concept, eight measurements of the resulting shape after the G-P mapping were used: size of range of points in each x, y, z axis; mean position of points in each of x, y, z axis; number of iterations of the G-P mapping algorithm; and the total number of positions visited.

D. Novelty Measure

Novelty search allows the trajectory of the evolution through the shape space to be controlled, with emphasis on exploring the space of all possible shapes. This allows the expressibility of the G-P mapping to be evaluated without reference to a particular application, which would be defined by a fitness function in a standard (possibly multi-objective) EA.

The novelty measure is based on the distance between each member of the population in the space defined by the shape metrics. Firstly, the K nearest neighbours are calculated using these distances. The novelty measure is then the mean of these distances. Expressed formally, for each individual x, the novelty score $\rho(x)$ is defined as:

$$\rho(x) = \frac{1}{K} \sum_{k=1}^{K} \delta(x, \mu_k) \tag{1}$$

where $\delta(x, \mu_k)$ is the Euclidean distance between individual phenotype x and its k^{th} nearest neighbour in the shape metric space.

This approach allows the search algorithm to identify genotypes that are on average the *furthest away* from their K *nearest* neighbours.

E. Mutation

The genomic representation is a tuple, so mutation must accommodate the different data types in each gene. Since the goal is to explore shapes, and the metrics use the growth positions to measure the shapes, only the growth genes were mutated (so the first group of genes in table I are fixed). Similarly, no genes are added or removed from a genome during mutation in order to get a clearer understanding of what shapes are possible from a genome of fixed size. A single

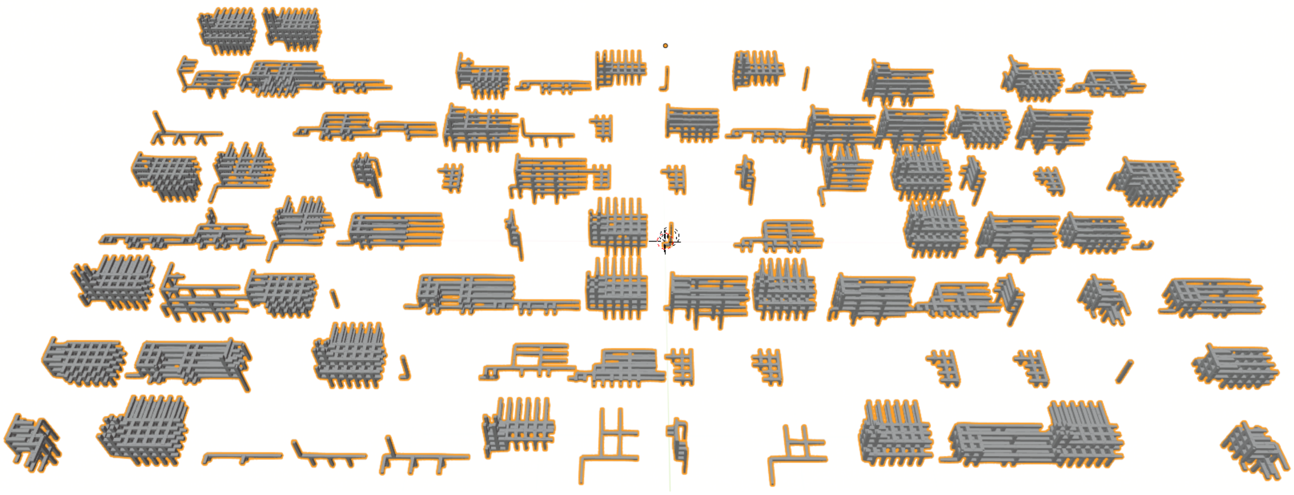

Fig. 5. State of the population of shapes after 100 generations using novelty search

mutation is defined as an increment or decrement of the integer value of the Shape, Start or Stop fields.

The dominance value can also be mutated, but the grouping stays fixed, and the new dominance value matches another group of genes, then that group adopts the original value of the group being mutated. For example, a mutation to the dominance value of gene 7 from 48 to 47 would change the dominance value of genes 7, 8 and 9 to 47, and genes 10, 11 and 12 to 48. Note that as in other Novelty Search applications, crossover is not used in this implementation.

F. Evolution

Apart from the novelty measure itself, well-established genetic algorithm techniques can be used pretty much arbitrarily for novelty search. To allow straightforward comparison related techniques, the evolutionary algorithm is also based on the findings of Gomes et al [8], as detailed in algorithm 2. Following Gomes, a population size of 100 was used. The population was initialised with single mutations of the seed shape described above. Each generation involves the following stages as described in algorithm 2:

1) obtain the shape metrics from each member of the population (SHAPEMEASURE function)
2) get the mean distance d to the $K = 15$ nearest neighbours of each member of the population (MEANKDIST function)
3) Elitism: rank the population by d - the individuals with the 5 largest d values are copied directly to the next generation.
4) Tournament selection with a tournament size of 2 is used to complete the individuals entered into the next generation. The winner of each tournament is then subjected to a mutation event (MUTATEGENE & WINNER functions).

III. Experiments

The experiments with these configurations were intended to determine whether novelty search was capable of exploring the range of available shapes that could be reached from a hand-designed initial shape using the configuration outlined above. Accordingly, three different configurations were run, each with a population size of 100 and running for 100 generations, resulting in 10,000 mutations in total per trial (excepting elites where used). The first of these acted as a baseline for the experiments. Here, each generation was produced by mutating each member of the previous generation. The result is a random walk through the mutation space of the genotype, whereby the distance of the corresponding phenotypes to the seed could be measured in the same manner as the other techniques.

The second experiment followed algorithm 2 exactly, closely following the recommendations in Gomes et al [8] for novelty search in evolution of effective maze-solving behaviours. Properties of the resulting phenotype shapes are captured using simple shape metrics to define the distance measure by which the novelty search was executed.

Finally, the third experiment examined the use of an archive of visited shapes to add a 'memory' to the stages in novelty search. If an archive is not available, it is possible that the distance measure could select mutations that caused previously observed states to be revisited. To investigate the effect of an archive the two most novel individuals at each generation were added to an archive of phenotypes. The archive was used along with the current generation to calculate the novelty distance metric for each generation. Other than that, the third experiment was identical to the second.

Ten trials of each configuration were used, each with different random number seeds to compensate for the influence of random events at particular points in the trial.

IV. Results

One of the challenges in investigating the range of shapes a mapping can produce is to determine how to measure success. There are two routes to this evaluation in this contribution. Firstly, the increase in the mean distance of each generation

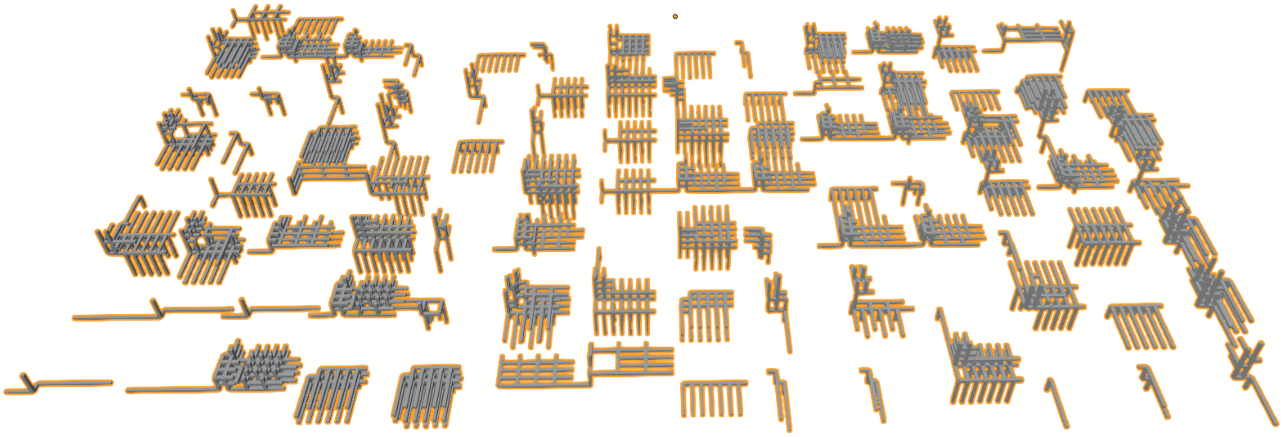

Fig. 6. State of the population of shapes after 100 generations using novelty search with archive

to the initial shape is studied. This is done for each of the three configurations as shown in figure 3. Secondly, the phenotypes in the final generation can be explored to get visual confirmation of the range of shapes that have evolved.

A. Quantitative analysis

Figure 3 shows clearly that evolution by novelty search generates phenotypes that have greater distance to the seed shape than simple random walk. Some other interesting features are revealed. The control experiment using random walk shows a gentle, regular increase in distance until around generation 50, when there is little further increase. The change in distance with each generation was similar for each trial in this configuration, as shown by the tight arrangement of individual lines around the mean. The novelty search with an archive is shown in red in the figure. There is more variability in each trial and generally the distance to the initial shape is around double the random walk. The novelty search configuration without an archive evolves on average to the greatest distance from the initial phenotype shape. This is unexpected since the lack of an archive means there is no ‘memory’ of shapes that the search algorithm is able to mutate away from.

B. Qualitative analysis

An example of the population at the end of one of the random walk trials is shown in figure 4. It is clear that after 100 mutations, little if anything of the table-like shape of the initial phenotype remains. The population is arranged on a grid, but some members of the population are not visible, meaning that there were no growth genes firing at the initial x, y, z position of [0,0,0]. This demonstrates that whilst growth iterates over a single position, then mutation can be highly deleterious on the phenotype. Also, many of the shapes that are present are less complex than the initial shape for the same reason: mutations have occurred which prevent any gene from firing early on in the G-P mapping.

By contrast, the final population of the novelty search trial without an archive is shown in figure 5. The shapes are generally more complex than the random walk although there are a small number of simpler shapes as well. One of the striking features is the manner in which the shapes fill their convex hull with relatively little branching of the growth.

The final population of one of the trials with an archive is shown in figure 6. Visual inspection indicates that the shapes tend to be less dense than those generated in novelty search without archive, with less regular filling along the orthogonal axes. Although this is difficult to measure in a statistically useful way in the current environment, this implies that the archive is having some positive effect on shape generation that isn’t currently detected with the shape measures.

V. Discussion

Novelty search has not been previously used in the area of generative design. It is intriguing to consider that generative growth has many similarities with maze-solving - both consider a range of positions at each stage in the iteration and both require an explicit method of making decisions based on local context.

The ability of novelty search to drive the exploration of the “reachable” shapes available from an initial seed has been demonstrated in this paper. Both configurations generated shapes that were more than double the distance from the seed than could be achieved by random walk. A counter-intuitive feature of the results is that the search using an archive yielded populations that were not as distant from the initial seed as the configuration without an archive. How could adding memory to the search make the exploration of the space less wide? One explanation of this is that features of the G-P mapping mean that there are critical points in the evolutionary trajectory that must be visited several times to ensure mutation along a pathway that yields shapes that are the most distant from the seed. Using an archive prevents revisiting previously novel shapes, forcing evolution down the first mutation pathway that emerges. An alternative explanation would be that the shape metrics used in these experiments are not capturing the richness of the search, so the archive-based search is continuously finding new features that are not rewarded in the distance measure. There is some empirical evidence for this in figures 5 and 6, where the pattern of branching between the two configurations is visually different.

It is also intriguing that the random walk experiment shows a gradual slowing down in the rate of increase in distance towards the end of each run. Why would random mutations uniformly drive the system to a constant distance from the seed? Figure 4 gives a clue as to why this might be - random mutations can be highly deleterious in terms of the growth of the shape, and if this happens to a genotype, further mutations are unlikely to chance upon richer phenotypes.

These observations indicate an important issue for generative design. The simple shape metrics used here are the only means of evaluating the phenotypes. The shape metrics are linear, whereas the G-P mapping is highly non-linear, so mutations in the genotype can have no effect or a huge effect, depending on the configuration of the rest of the genotype. Thus, more sophisticated ways of evaluating phenotypes are required.

The genomic representation used in these experiments has a mix of grammar-like properties combined with position-based switching of particular genes. This approach is useful because if particular features are needed in precise positions, this can be accommodated in the representation. The Dominance field in the gene tuple facilitates this, but may also be the cause of highly deleterious mutations. A further challenge remains in designing a representation that has these facilities, but is also more capable of generating useful and robust phenotypes under mutation. The framework presented here provides a clear mechanism for evaluating these developments.

Generative design is intended to open up the design space, so with this genotype and the novelty algorithm in this work, the design alternatives and the multitude of phenotypes that can arise from this single genotype can be explored. However to ensure the phenotype has manufacturing credibility we must integrate manufacturing based constraints within the novelty measurement. In product development these constraints are dictated by multidisciplinary analysis, design for manufacture, requirements, goals and quality. Each of these constraints can have their own effect on the products shape and form, determining the final design structure. The inclusion of such constraints into this work, for example a common finite element constraint such as Stress, will directly influence how the genotype evolves in the design space thus radically altering the resulting phenotypes.

Novelty search is an important tool in generating diversity in the design space. Future work will focus on methods for adding environmental feedback to the search, and the inclusion of specific fitness measures to drive the later stages of evolution towards a range of applicable solutions.

The software used to generate the analysis in this contribution was written in R and is available from the authors on request.

References

[1] V. Singh and N. Gu, "Towards an integrated generative design framework," *Design Studies*, vol. 33, no. 2, pp. 185 – 207, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0142694X11000391

[2] M. Albert, "Generating designs tor manufacturability." *Modern Machine Shop*, vol. 91, no. 9, pp. 78 – 85, 2019.

[3] W. Zhang, M. Price, T. Robinson, D. Nolan, P. Kilpatrick, and S. Barbhuiya, "Gene-inspired development of innovative design: principles and algorithm," submitted for review in Procedia CIRP.

[4] J. Lehman and K. O. Stanley, "Exploiting open-endedness to solve problems through the search for novelty," in *ALIFE*. pdfs.semanticscholar.org, 2008, pp. 329–336. [Online]. Available: https://pdfs.semanticscholar.org/8cb8/3e8ad10fed42f24932cead992b4ae9ba1625.pdf

[5] W. Zhang, M. Price, T. Robinson, D. Nolan, D. Nikolopoulos, S. Barbhuiya, and S. Kyle, "Design gene representations for emergent innovative design," in *17th International Conference on Manufacturing Research ICMR 2019*. IOS Press, Sep. 2019. [Online]. Available: https://pure.qub.ac.uk/portal/en/publications/design-gene-representations-for-emergent-innovative-design(c02006bc-76cf-43d9-abe8-d8e7261175a9).html

[6] P. Prusinkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants*. Springer Science & Business Media, Dec. 2012. [Online]. Available: https://play.google.com/store/books/details?id=4F7lBwAAQBAJ

[7] S. Kistemaker and S. Whiteson, "Critical factors in the performance of novelty search," in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '11. New York, NY, USA: ACM, 2011, pp. 965–972. [Online]. Available: http://doi.acm.org/10.1145/2001576.2001708

[8] J. Gomes, P. Mariano, and A. L. Christensen, "Devising effective novelty search algorithms: A comprehensive empirical study," in *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '15. New York, NY, USA: ACM, 2015, pp. 943–950. [Online]. Available: http://doi.acm.org/10.1145/2739480.2754736