Knowledge-Based Repair for Knowledge-Lean Techniques in Non-Routine Design

Michael A. Rosenman and Nicholas Preema

Abstract— This paper presents a methodology for producing good design solutions more efficiently. The methodology is based on augmenting a conventional evolutionary design approach with a method for improving suboptimal design solutions with a domain-specific knowledge-rich approach. This approach is based conceptually on the practice of plastic surgery, i.e. making minor adjustments to an entity, based on some desired qualities, i.e. specified fitness function. Additionally, the modifications made to the phenotype may require the re-engineering of the genotype to accord with the modified phenotype if the entity is to be used further in evolutionary operations. A method for genotype re-engineering is proposed in the domain of cellular growth generation.

Keywords— evolutionary design, genetic algorithms, plastic surgery, genetic re-engineering.

I. Introduction

NON-ROUTINE design tasks are characterized by the lack of knowledge available for their immediate solution [3]. Thus knowledge-lean approaches, such as evolutionary computation methods, are well suited to the task of non-routine design [14] [1], [2], [7]. Evolutionary computation methods are able to arrive at reasonable solutions fairly quickly to begin with but then need many generations to make subsequent small improvements, [3], [11]. A great deal of effort can be expended to make a small (but maybe critical) improvement. In general, there is no guarantee that such an improvement will be found. In addition, in non-routine design, it is not always possible to perfectly specify the fitness function such that optimal solutions will be found since the design task is not well-known. This paper argues that, even in such conditions, it is possible to obtain reasonable solutions within the bounds given and then, using these resulting solutions as a guide, make improvements to obtain better solutions. For example, Fig. 1 (a) shows a configuration of 12 cells that may arise after a number of generations.

To produce an improvement such that the protrusion is removed and the indentation is filled (leading to a square in this case), Fig. 1 (b), may take a great deal effort from the evolutionary computation process. Where the number of cells is large, (>=100), the ability of such a system to remove such irregularities will be inefficient. While we can see that removing the protrusion and filling in the indentation would lead to a good solution, the evolutionary system based on random genetic operations (crossover and mutation) on the genotype may not be able to produce the required solution within a feasible timescale.

(a) an almost 'perfect' solution (b) an improved solution

Fig 1. Improvement of a design solution.

Plastic surgery is a practice whereby features of an entity (generally a human) are altered to improve the appearance of that entity. This may be for cosmetic purpose or for more serious reasons. In all cases, the effect is on the entity itself, i.e. the phenotype, and there is no change to the genotype (DNA). Since evaluation is done on the phenotype, any improvement to the phenotype gives the entity a better chance of survival or attaining its goal, e.g. attaining self esteem, attracting other entities, etc. Plastic surgery is generally done to correct minor features; an entity has been generated in some way, but is defective in some features and minor corrections are made (to the phenotype) to improve it. It can be seen that specialized knowledge is required to modify the phenotype. Different domains require different knowledge. The phenotype and the defects must be recognized and the means for modification determined and implemented.

As in the human example, any modification to the phenotype (design solution) is not transmitted to the genotype. Any 'children' may carry the defective genes and reproduce the same defects. However, in design, if the modified design solution is the final solution required, and no more processing is to take place, then this does not matter as the genotype was just the means to the end and is no longer of any interest. However, if the modified design solution is only a part solution and is required to take part in further evaluation, a problem exists since all evolutionary operations are carried out on the genotype. In that case, its genotype must be re-engineered to match the modified phenotype. This difficult to implement since, in general, there is no known connection between the phenotype and the genes in the genotype.

Manuscript received February 14, 2008.

M.A. Rosenman is with the Key Centre of Design Computing and Cognition, The University of Sydney, Australia (phone: +61 (2) 9351; fax: 303-555-5555; e-mail: mike@arch.usyd.edu.au).

N. Preema, is also with the Key Centre of Design Computing and Cognition, The University of Sydney, Australia.

Although there are some examples of genetic engineering [10], in general, there is no universal knowledge on how to modify the genotype to produce required characteristics of the phenotype.

A conventional evolutionary method may be used, in situations where the form required is not known a priori, to produce possible solutions which are reasonably good but need some improvement. Once the solutions are produced, any suitable solutions can be improved by making small modifications to the phenotype. If the resulting solution is required to take part in further evolutionary operations, genetic-re-engineering is required as described above. This paper will put forward a method for re-engineering the genotype in a given design representation.

II. Evolutionary Design

While knowledge-lean methods, such as evolutionary design, are good for discovering possible reasonable solutions where little knowledge is known a priori regarding the form of the solution, they are generally computationally expensive and may not be able to make the necessary improvements in a reasonable time with reasonable resources. Additionally, in an environment where there exists little a priori knowledge, it is not always possible to perfectly specify the requirements, i.e. formulate a ‘perfect’ fitness function. On the other hand, while knowledge-rich approaches can solve problems where the problem is well defined and the knowledge and methods required are also known, they operate in specific problem areas and, even within those areas, have little capacity for producing innovative solutions.

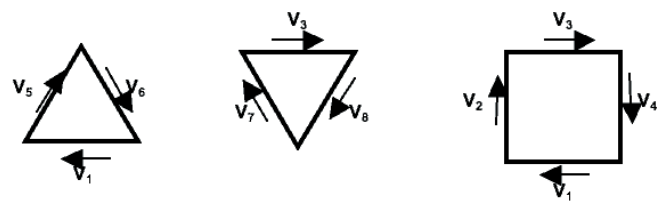
This paper presents an approach combines knowledge-lean and knowledge-rich approaches to increase the efficiency of producing good design solutions in a non-routine design problem environment. The conventional evolutionary computation approach generates reasonably good solutions within given initial specifications and the proposed plastic surgery makes small modifications as necessary based on local knowledge of the problem once the solutions are evident.

A. A Design Representation

In general, design can be defined as the derivation of structure (form) that will satisfy a given set of requirements [18]. In its simplest mode, the construction of form can be thought of as the set of decisions for locating a set of cells of substances, where a substance may be physical (composed of a physical material) or virtual (e.g. composed of graphic entities or pixels). The construction of a spatial entity may be considered as the allocation of a physical substance composed of a ‘space’ material, i.e. a number of space cells. So, in summary, design can be see as the generation of form which can be produced by cellular growth. In an evolutionary design approach, a gene selects a module of substance and allocates it to some location. In the approach of Rosenman [14], [15], a gene locates a module of substance relative to another module. A gene, GN, is thus (M_1, M_2, L_{12}) where M_1 and M_2 are two modules of some substance and L_{12} is the operator for locating module M_2 relative to module M_1. A module, M_i, may be a single unit cell or a set of unit cells already grouped and, in general, M_1 and M_2 need not be composed of the same substance. Nor, in general, does a grouping of units necessarily need be constructed of units of the same substance. The instructions for a complete design solution, i.e. a genotype, G, is a sequence of genes where G = (GN_1, … GN_m) and GN_i, i = 1,m is a gene.

In the approach, based on the joining of polygons representing units of space, the allocation operation is founded on the joining of polygons through their free edges represented as vectors. Fig. 2 shows the representation of two triangles, T1 and T2, and a square, SQ1, as closed vector loops. The vector V1 is a vector of length 1 unit and angle 180^o, the vector V2 is a vector of length 1 unit and angle 90^o, V5 is a vector of length 1 unit and angle 60^o, etc. The phenotype, P, of each polygon is given as the loop of vectors. Since this is a loop, the start point of the loop is immaterial, although in the examples it is given as the lowest-leftmost point.

Polygons may be joined by conjoining counteractive (equal and opposite) vectors [13].



P = (V5, V6, V1,) P = (V7, V3, V8,) P= (V2, V3, V4, V1,)

a) T1 b) T2 c) SQ1

When a number of squares are joined randomly the resulting shapes (polyminos) are not likely to show much regularity, especially if the number of squares is large. A shape with many protrusions and indentations will have many changes of direction on its boundary. Fig. 4 shows 40 random generations of 16-unit polyminos.

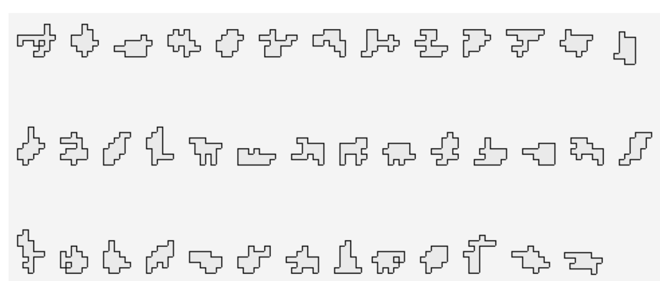


Figure 3. 40 random generations of 16-unit polyminos

For the example in Fig. 3, using 16 cells, the evolutionary program will evolve the shapes to find L- and T-shapes as well as rectangles and the 4 x 4 square. However, when using a small-scale cell to allow for small discriminations in the dimensions, the number of total cells will be very large. For example if the scale of the cells in Fig. 3 were reduced by a factor of 10, allowing for increments in length of say 10cms rather than 1m, the total number of cells would be 1600. For a large number of cells, an evolutionary process, will, after a number of generations, give some indication of possible satisfactory shapes but will, usually, not be able to perfectly smooth out all the protrusions and indentations. One reason for this may be that the fitness function is not perfectly specified since such knowledge may be beyond the current knowledge of the design situation. For example, a fitness function based on minimizing the perimeter to area ratio will find that, with a large number of cells, the number of cells at the perimeter is small compared to the number of interior cells which are fairly well compacted. Thus most of the solutions will show a fairly high score for that fitness function. The process will not be able to make any significant improvement in any reasonable time. Fig. 4 shows a shape of 85 cells after evolution over a number of generations. The perimeter to area ratio shows a fitness of 85.8% compared to the ideal of a square of area 85 (perimeter = 36.88).

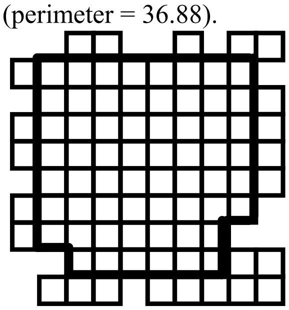


Figure 4. A polymino of 85 units

The reason for the high fitness, even though the perimeter is not very smooth, is that the central part of the shape (outlined in bold), which contains the majority of the units, is quite compact. Continuing the evolutionary process usually leads to convergence on one or other of the ‘better’ solutions to date before any major improvement is attained.

III. Plastic Surgery in Evolutionary Design

In an evolutionary system, selection acts with respect to the phenotype. Those members whose phenotypes are judged to be well-suited to their environment will have a better chance of survival and of propagating their genes [7]. Thus any improvement in the phenotype, regardless of any change in the genotype, will improve that member’s chance of survival and propagation. Of course, this improvement will not be transmitted to the member’s descendants. In a design domain the fitness of the design is what counts, how it got to be that way is secondary.

The Merriam-Webster dictionary [9] states that plastic surgery is:

“**:** surgery concerned with the repair, restoration, or improvement of lost, injured, defective, or misshapen body parts”

Plastic surgery is aimed at improving the organism’s survival in its environment (whatever survival may mean).

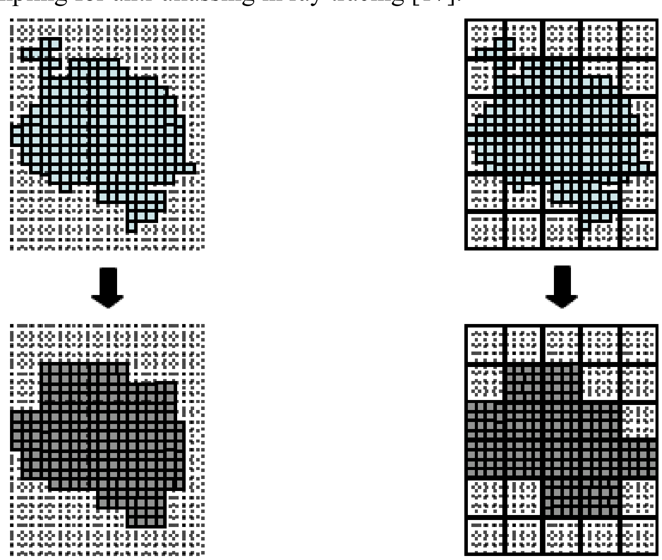
Plastic surgery is proposed here as a solution to improving a phenotype (design solution) generated through an evolutionary computation method. It is proposed as a general concept where the issues are as follows:

- design solutions are produced which are reasonable but could be improved by relatively small modifications. This requires particular domain knowledge.
- if the design solutions are the end product of the process then the modified phenotype is the final solution and it is no longer necessary to consider the genotype. However, if the ‘solution’ is part of an on-going process, e.g. a component of a hierarchical composition, it may be necessary to re-engineer the genotype to match it to the new phenotype so as to enable it to be operated on by the genetic operators.
- the modifications should be limited to relatively small remedial improvements. It is not meant to carry out major reconstructions of the phenotype as this leads to too large a departure from the solutions found.

While an example in the domain of the generation of smooth polygons will be used to demonstrate the concepts, this paper suggests that the general principles of plastic surgery and genetic re-engineering could be applied to all domains since all design is a function of locating elements in a certain configuration.

IV. Methodology

The implementation of plastic surgery consists of several transformation functions. There exist various smoothing algorithms mainly in image processing, where they are used to produce smoothed surfaces from polygonal or noisy surfaces [5], [6], [19]. Algorithms such as Potrace [12] transform bitmap images into vector graphics. Another process uses sampling for anti-aliassing in ray tracing [17].



(a) smoothing maintaining sampling

(b) smoothing using smaller unit size

Figure 5. Comparing plastic surgery to sampling