

Reinventing the wheel, or evolving to the wheel

Luis Alberto Gómez¹
Francisco Antonio Pereira Fialho²

Abstract:

Genetic Algorithms is based on the assumption that nature can teach us how to develop artifacts able to comply with restrictions imposed by the environments. Those artifacts can be concrete or mental. In this article the authors propose the concept of Geometric Genetic Programming (GGP), which opens a wide variety of opportunities for applying the concepts of Genetic Programming (GP) to engineering problems. This papers shows how it is possible to evolve to well known, or even completely new designs starting just with a set of physical laws and a fitness function. To show how GGP works the challenge is to produce the most common machine in nature, the wheel.

Key Words: Knowledge Engineering. Geometric Genetic Programming.

¹Dr. Professor of the Departamento de Engenharia Civil da UFSC, lagomez@gmail.com

²Dr. Professor of the Departamento de Engenharia do Conhecimento da UFSC, fapfialho@gmail.com

1 INTRODUCTION

In their 2003 Scientific American article Koza et al suggest that the evolution theory can be use to reproduce technical inventions or even to create inventions. According to Koza:

One of the practical commercial areas for genetic programming is design. In essence, design is what engineers do eight hours a day and is what evolution does. Design is especially well suited to genetic programming because it presents tough problems for which people seek solutions that are very good but not mathematically perfect. Generally there are complex trade-offs between competing considerations, and the best balance among the various factors is difficult to foresee. Design usually involves discovering topological arrangements of things (as opposed to merely optimizing a set of numbers), a task that genetic programming is very good at. (KOZA, 2003, p. 56)

The proposed idea ofgeometric genetic programming has a lot to do with topological arrangement. Cell elements combined together, conforming a shape or solid. From an initial population of randomly generated shapes or solids, the Geometric Genetic Programming(GGP) algorithm will choose the best shapes or solids according to a given criteria defined in a function to be minimized. It is expected that newer generations behave better than previous in relation to the function.

2 GEOMETRIC GENETIC PROGRAMMING

The genetic algorithm will evolve the final shape, defining whether a given cell of the body should be present or not and the characteristics of the cells, depending on the fitness function.

The idea is to define a body that can be either 2D or 3D, composed by a given number of cells, for the sake of simplicity, the body will be represented by a matrix.

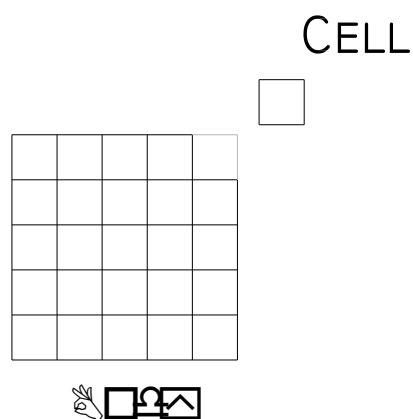


Figure 1. 2D body

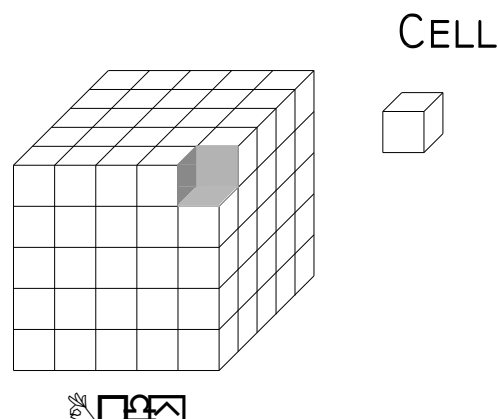


Figure 2. . 3D body

Even that the body is 2D or 3D, each cell may have a higher order dimension. The new dimensions may represent physical characteristic of each cell (Young's module, weight, electric resistance, magnetic permeability etc.) or other intrinsic characteristic (price, for example).

The fitness function, to be minimized or maximized, should be dependent on the geometric characteristics of the shape (that changes from generation to generation). It can be the response of the shape to forces and momentum, the speed for transmitting heat etc.

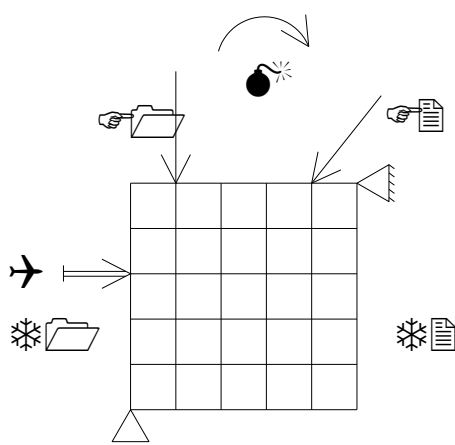


Figure 3. Example of fitness function

As it is clear the larger is number of cell, more accurate will be the shape, but the complexity of the problem grows exponentially. As example in a 2D 5x5 matrix just considering two possibilities for each cell (whether the cell exist or not) it results in 2 to the power 25 possibilities, difficult enough to be tackled by conventional algorithms. The number of possibilities can be reduced substantially if the program allows some kind of symmetry.

3 REINVENTING THE WHEEL – SETTING UP THE EXPERIMENT

For developing the experiment, we assume a 5x5 matrix, each cell in the matrix, can be either empty or full according the shape to be analyzed. For the 5x5 matrix there is the possibility of 2 to the power 25 different possible shapes. Figure 4 shows the matrix and a shape of a hollow wheel defined by the marked cells.

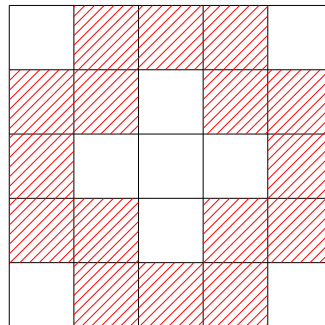


Figure 4. Matrix that defines the shape. Hatched cells are full and blank cells are empty

As we would like complete freedom to the program we will not restrict to symmetrical shapes.

4 CROSS-OVER

The cross over will be performed as shown in figure 5, taken parts of two shapes to be crossed.

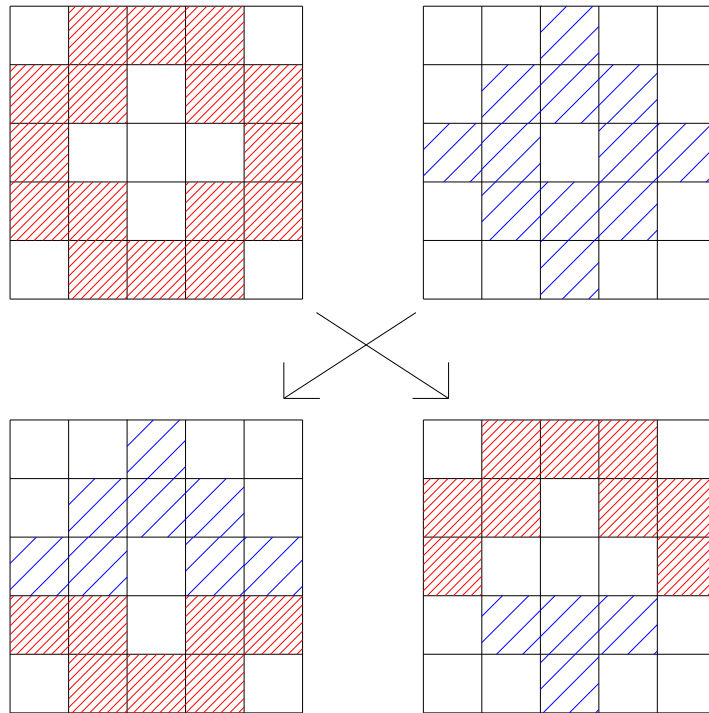


Figure 5. Cross-over of the shape

5 MUTATION

Mutation takes place by adding or eliminating a random cell (fig 6)

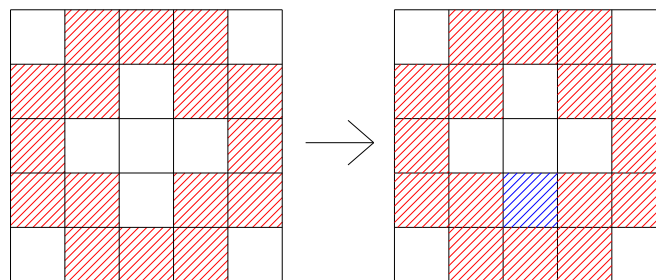


Figure 6. Mutation of the shape

6 FITNESS FUNCTION

Force is always applied at the center of gravity of the shape, simulating the action of an axis. It is assumed infinite friction among the shape and the surface.

The function to be minimized is associated the force necessary to over turn the shape, in our case the force necessary to spin the wheel. This happens when the resultant force drops out of the base of the shape. Figure 7 shows the three possibilities. In Fig 7a the force is small and the shape in stable balance, as the force becomes larger (Fig 7b) the shape passes to an instable balance state and in fig 7c the force makes the shape to rotate.

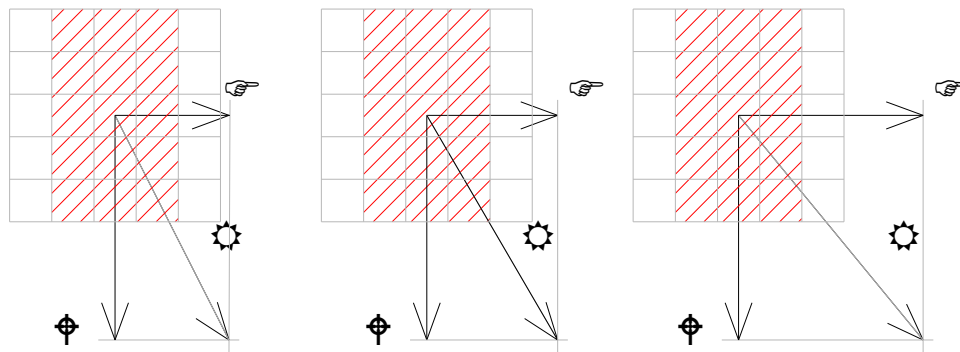


Figure 7. As the force increases the shape passes from stable balance to unstable balance and finally spins

We will call limit force the force necessary to reach the unstable balance.

As the shape rotate, the size and the topology of the base change, so does the limit force necessary to spin the shape (Fig. 8)

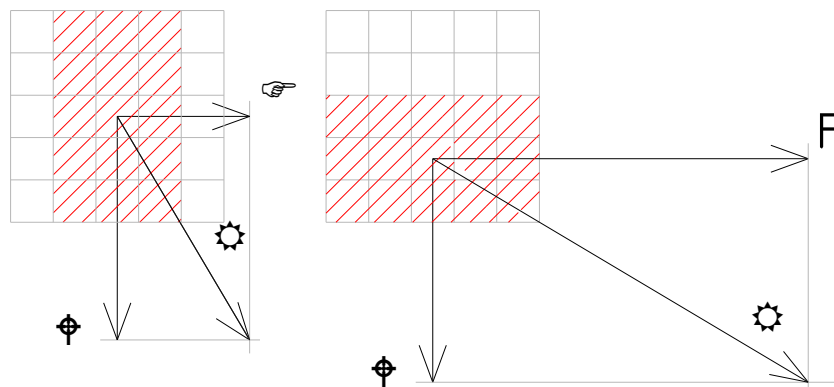


Figure 8. The limit force depends on the base of the shape

In order to normalize the function the limit force is divided by the “weight” of the shape (number of marked cells). F can be obtained graphically and instead of N its unit will be a number and fraction of cells)

The function to be minimized by the genetic algorithm is:

$$\Phi = \sum_{i=1}^n r_{i_i}^{\text{lim}} / W$$

where n is the number of steps to make a complete turn, in our case is 20 (5 cells times 4 sides)

7 DRAWBACKS

The use of a matrix of 5x5 has the disadvantage of not allowing smooth shapes, as well as making difficult the operation of rotating the wheel. If computational time is not a problem, larger matrixes can be used.

8 CONCLUSIONS

The geometric genetic programming proves to be a novel way to create shapes and volumes that respond better than others to a given set of restrictions.

Although the example in this paper works with shapes, the proposed method can easily be transported to a 3D by using cubes instead of squares, allowing to work with solids.

In this papers we develop shapes by working with geometry (the fitness function was defined exclusively on the geometric characteristic of the shape), but there is no such limitations, finite elements methods can be used in the fitness function, the shape or solid can be defined with their physical properties (limit stress, elasticity modules etc.)

Reinventando a roda, ou evoluindo até a roda

Resumo:

Algoritmos Genéticos são baseados na premissa de que a natureza pode nos ensinar a desenvolver artefatos capazes de satisfazer restrições impostas pelo ambiente. Estes artefatos podem ser concretos ou mentais. Neste artigo os autores propõem o conceito de Programação Genética Geométrica (PGG), abrindo uma ampla variedade de oportunidades para a aplicação dos conceitos da Programação Genética para problemas de engenharia. O artigo mostra como é possível evoluir para designs bem conhecidos e inovadores a partir de um conjunto de leis físicas e da adequação aos objetivos pretendidos, definidos por meio de uma “fitness function”. Para exemplificar o funcionamento da PGG, usamos como exemplo, a máquina mais comum na natureza, a roda.

Palavras-chaves: Engenharia do Conhecimento. Programação Genética Geométrica.

REFERENCES

Koza, J., Keane, M. Streeter, M. **Evolving Inventions**. Scientific American 2003 pp52- 59

Koza, J. ,**Genetic Programming: On the Programming of Computers by Means of Natural Selection**.. MIT Press, 1992.