Generative Design - Genetic Algorithm applied to Structural Beam Design

Final Project - Artificial Intelligence - DCC/UFMG - 2018

Alvaro Salgado
Bioinformatics/UFMG
Belo Horizonte, Minas Gerais
alvarosalgado@ufmg.br

Gustavo Scarabelli
Civil Engineering/UFMG
Belo Horizonte, Minas Gerais
gustavo.scarabelli@hotmail.com

ABSTRACT

Generative design is an exploration and optimization approach to design. This work applies one of its techniques, genetic algorithms, to the design of a structural beam.

CCS CONCEPTS

• **Applied computing → Computer-aided design**;

KEYWORDS

Generative Design, Genetic Algorithm, CAD, Optimization

ACM Reference Format:

Alvaro Salgado and Gustavo Scarabelli. 2018. Generative Design - Genetic Algorithm applied to Structural Beam Design: Final Project - Artificial Intelligence - DCC/UFMG - 2018. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The present work is part of a final project in the Artificial Intelligence class at DCC-UFMG, 2018/01. The main objective is to apply one of the methods studied to a real world case, inserting it in a larger context.

The theme chosen was "Generative Design" and the use of Genetic Algorithms as a means of design exploration and optimization. Generative Design is the use of computational tools in the design process, which is based on much experimentation and prototyping. This way, the solution space can be explored, new ones uncovered and improved.

Genetic algorithms, emulating nature's design process, can be seen as both exploration and optimization. It is one of many techniques used in GD, and we will use it here to design a structural beam under certain constraints.

This work was developed in collaboration with undergraduate Civil Engineering student Gustavo Scarabelli.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 GENERATIVE DESIGN

"Generative design is the automated algorithmic combination of goals and constraints to ***reveal*** *solution".*(https://www.linkedin.com/pulse/what-generative-design-anthony-hauck/)

Design creativity techniques encourage divergent thinking and the main incentives for adopting generative design (GD) are to use computational capabilities to support human designers in the design process. The ability to explore larger design space and support design generation is one of the main objectives. [5] According to [5], there is a general consensus that design is a co-evolutionary process. Often the designer starts with an ill-defined problem and as the design activity continues the problem and solutions co-evolve and mutually guide each other. During this process, the designer's view of the problem and solution is dynamic and changes, as the designer interacts with the task. If the designer's present state (of perception, experience) can be altered it is possible to create greater variety in the solutions generated. Digital design approaches can facilitate greater design exploration through different viewpoints.

The recent developments in additive manufacturing are another driving force behind GD, as it now becomes possible to manufacture solutions that would be impossible with more traditional manufacturing methods.

The design of complex artifacts such as buildings or products requires designers to explore multiple alternatives. Currently, most of this exploration happens at the conceptual stage of design with the aid of pencil and paper. CAD is rarely used at this stage of the design process. In its current form, CAD is a tool of implementation and increasingly a tool of analysis that is most useful at the later stages of the design process. But at this later stage, all the important commitments have already been made and significant improvements cannot be made. Generative design is largely about using computer aided design in the initial stages of the design process.[2]

Some of the most commonly used GD techniques are: shape grammars (SG), L-systems (LS), cellular automata (CA), genetic algorithms (GA) and swarm intelligence (SI).[5]

(1) Cellular automata (CA): is a collection of cells on a grid of a specified shape that evolve over time according to a set of rules driven by the state of the neighboring cells.
(2) Genetic algorithms (GA): GAs and genetic programming are evolutionary techniques inspired by natural evolutionary processes. GAs use the analogues of evolutionary operators on a population of states in a search space to find those states that optimize a fitness function.

(3) Shape grammars (SG): A Shape Grammar is a set of shape rules that can be applied to generate a set or language of designs.
(4) L-systems (LS): LSs are mathematical algorithms known for generating fractal-like forms with self-similarity that exhibit the characteristics of biological growth. LSs have been used for a wide range of design problems from simple computer graphics patterns to complex city planning and simulation.
(5) Swarm intelligence (SI) and multi-agent societies: Agent based models (ABM) are often used to implement social or collective behaviors. SI is the property of a system whereby the collective behaviors of unsophisticated agents interacting locally with their environment cause coherent functional global patterns to emerge.

Amongst the many benefits of using a Generative Design approach are the exploration of previously unseen solutions coupled with optimization properties. Such solutions could lead to lower environmental impacts and higher sustainability characteristics along with a reduction of costs, by means of reducing material use and improving key attributes related to energy efficiency and positive social impact.

Of all the Generative Design techniques highlighted above, Genetic Algorithms are especially suited for optimization problems. Some of its advantages are:

- Regular design evaluation and improvement
- Multiple solutions
- Optimization
- Disruptive innovation

The challenge in applying GAs is finding the proper way to model the problem to be tackled, in terms of "phenotype", "genes", "fitness", "crossover" and "mutation".

3 GENETIC ALGORITHM

GAs are a heuristic solution-search or optimization technique, originally motivated by the Darwinian principle of evolution through (genetic) selection.

Evolution via natural selection of a randomly chosen population of individuals can be thought of as a search through the space of possible chromosome values. In that sense, an evolutionary algorithm (EA) is a stochastic search for an optimal solution to a given problem.[1]

A GA uses a highly abstract version of evolutionary processes to evolve solutions to given problems. Each GA operates on a population of artificial chromosomes. These are strings in a finite alphabet (usually binary). Each chromosome represents a solution to a problem and has a fitness, a real number which is a measure of how good a solution it is to the particular problem. Starting with a randomly generated population of chromosomes, a GA carries out a process of fitness-based selection and recombination to produce a successor population, the next generation. During re-combination, parent chromosomes are selected and their genetic material is recombined to produce child chromosomes. These then pass into the successor population.

As this process is iterated, a sequence of successive generations evolves and the average fitness of the chromosomes tends to increase until some stopping criterion is reached.

In this way, a GA "evolves" a best solution to a given problem.[3] According to [3], there are many choices that have to be made in designing a GA for a given application. The choice of encoding will depend on the nature of the problem. Non bit-string representations are now common, and include sequences of integer or floating point values. As the size of the allele set expands, for example, where the strings consist of floating point numbers, the set of possible chromosomes becomes considerably greater. Many modern (or nonclassical) GAs use a range of representational approaches to ensure that the set of possible chromosomes is a close match for the set of possible solutions of the problem. Having selected an encoding, there are many other choices to make. These include:

- the form of the fitness function;
- population size;
- crossover and mutation operators and their respective rates;
- evolutionary scheme to be applied;
- appropriate stopping/re-start conditions.

The usual design approach is a combination of experience, problem-specific modeling and experimentation with different evolution schemes and other parameters.

Genetic algorithms work best when schemata, parts of the genome containing atomic information, correspond to meaningful components of a solution. For example, if the string is a representation of an antenna, then the schemata may represent components of the antenna, such as reflectors and deflectors.[4]

4 STRUCTURAL BEAM DESIGN

Our work will use GAs to help designing a simple steel reinforced concrete beam. The problem was modeled to be visualized in a simple 2D solution space, where each dimension is one of the beam's cross section dimensions (base and height), the fitness function is the beam's cost, and the restriction is a minimum breaking bending moment.

Design of a steel reinforced concrete beam subjected to simple normal bending:

Data:

- Steel (CA50)$\rightarrow f_{yk} = 50\,kN/cm^2 \rightarrow f_{yd} = \frac{50}{1.15} = 43.478\,kN/cm^2$
- Concrete $\rightarrow f_{ck} = 20\,MPa \rightarrow f_c = 0.85\,\frac{2}{1.4} = 1.214\,kN/cm^2$
- Diminution coefficients: $\gamma_c = 1.4$, $\gamma_s = 1.15$, $\gamma_f = 1.4$
- $\epsilon_{cu} = 3.5‰$; $\epsilon_{su} = 10‰$ (Domain 3)
- $d'' = 5\ cm$

Neutral line calculations:

$$\frac{\epsilon_{cu}}{x} = \frac{\epsilon_{cu} + \epsilon_{su}}{d} \rightarrow x = \frac{3.5\ d}{13.5} \ \therefore\ x = 0.259 \cdot d\ [cm]$$

R_{CC} *and* R_{ST} calculations:

$$R_{CC} = f_c\ b\ y = f_c\ b\ \lambda\ x = 1.214 \times b \times 0.8 \times 0.259 \times d$$

$$\therefore\ R_{CC} = 0.252\ b\ d\ [kN]$$

$$R_{ST} = \sigma_s\ A_s\ = f_{yd}\ A_s\ \therefore\ R_{ST} = 43.478\ A_s\ [kN]$$

Calculation of breaking bending moment (ELU):

$$M = \frac{M_d}{\gamma_f}\ \therefore M = \frac{1}{1.4}\left[R_{CC}\left(\frac{h}{2} - \frac{y}{2}\right) + R_{ST}\left(\frac{h}{2} - d''\right)\right]\ [kN \cdot cm]$$

$$\sum F_H = 0 \therefore R_{CC} = R_{ST}$$

$$0.252\ b\ d\ = 43.478\ A_s\ \therefore\ A_s = 5.79 \cdot 10^{-3}\ b\ d\ [cm^2]$$

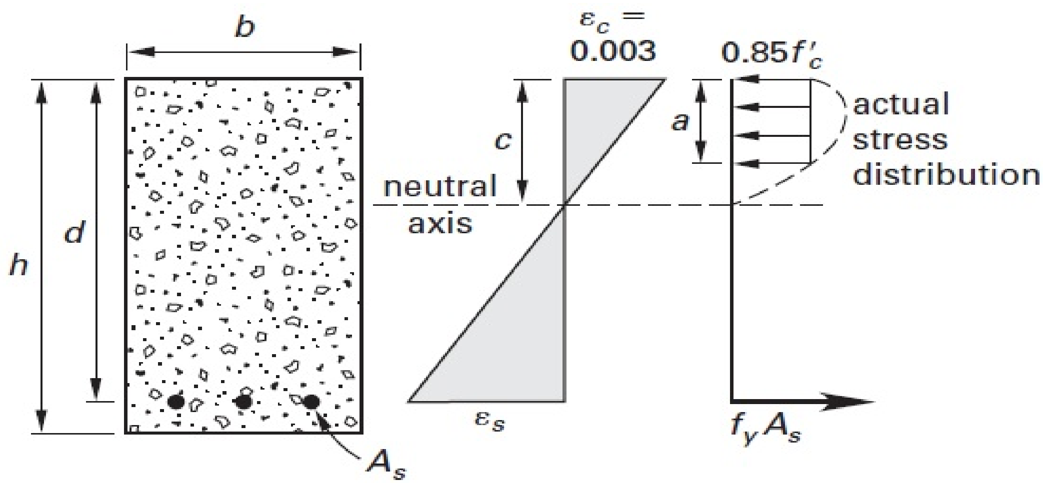

Figure 1: Beam stress (http://civilengineeringbible.com)

5 IMPLEMENTATION

Since our objective is to use GAs to design a structural beam, the first step is to understand what are the variables of interest and how to model them in a way suitable for GAs to deal with. In the previous section we showed how a simple beam is mathematically modeled, now we will translate it to GAs terms.

- Individual: each beam is an individual (a possible solution) whose main attributes are:
  - dimensions (b and h);
  - steel cross sectional area (A_s);
  - breaking bending moment (M);
  - cost.
- Genome: the genome is composed of two real values, b and h.
- Fitness: the fitness is each individual's cost, which should be minimized.
- Crossover: two individuals are combined to generate a new one by choosing, for each dimension (each gene), a random value in the range spanned by both parents for that dimension. For example, if one parent has $b = 20$ and the other has $b = 50$, the offspring will have $20 < b < 50$. Parents are randomly assigned.
- Mutation: whenever an individual is mutated, one of its dimensions is changed to a random valid value.
- Selection: at each generation, a simple "roulette" elimination is applied, maintaining the population size constant.

6 RESULTS AND DISCUSSION

Shown below is a result for a population of 100 individuals and a maximum of 50 generations. We defined a minimum breaking moment of 2000kN. The blue dots are the initial population, the red ones are the final population. The yellow dot is the best solution achieved. Since this is a very simple problem, it is easy to infer just by looking at the image where the global optimum is, where the 2000kN isoline crosses the lower boundary (minimum acceptable b is equal to 20cm).

Our tries didn't converge exactly to the global minimum. Probably because of the crossover scheme used, where all child individuals of a population, in each generation, will inevitably fall inside the region enclosed by the limits spanned by the population. Therefore, unless a mutation moves an individual outside this envelope, the population has no hopes of reaching anywhere outside of it.

One solution to this problem could be to modify the crossover function to generate children in an area around the parents, not exactly on the line connecting them. Also, the coupling of a second optimization algorithm is common practice, where the GA performs the initial exploration and converges to a region close to a possible solution, after which something like simulated annealing finishes the job.

7 CONCLUSIONS

Generative Design is a vast and interesting field. We applied one of its techniques to a simple problem as a way to solidify our knowledge in the basics of Genetic Algorithms and instill interest in the area on the undergraduate student under my co-supervision. It provided an engaging opportunity to discuss approaches to model and solve a problem computationally, an enriching experience for both of us.

REFERENCES

[1] Andries P. Engelbrecht. 2007. *Computational intelligence: an introduction* (2nd ed ed.). John Wiley & Sons, Chichester, England ; Hoboken, NJ. OCLC: ocn133465571.
[2] Sivam Krish. 2011. A practical generative design method. *Computer-Aided Design* 43, 1 (2011), 88–100.
[3] John McCall. 2005. Genetic algorithms for modelling and optimisation. *J. Comput. Appl. Math.* 184, 1 (2005), 205–222.
[4] Stuart J Russell and Peter Norvig. 2002. *Artificial intelligence: a modern approach (International Edition)*. {Pearson US Imports & PHIPEs}.
[5] Vishal Singh and Ning Gu. 2012. Towards an integrated generative design framework. *Design studies* 33, 2 (2012), 185–207.

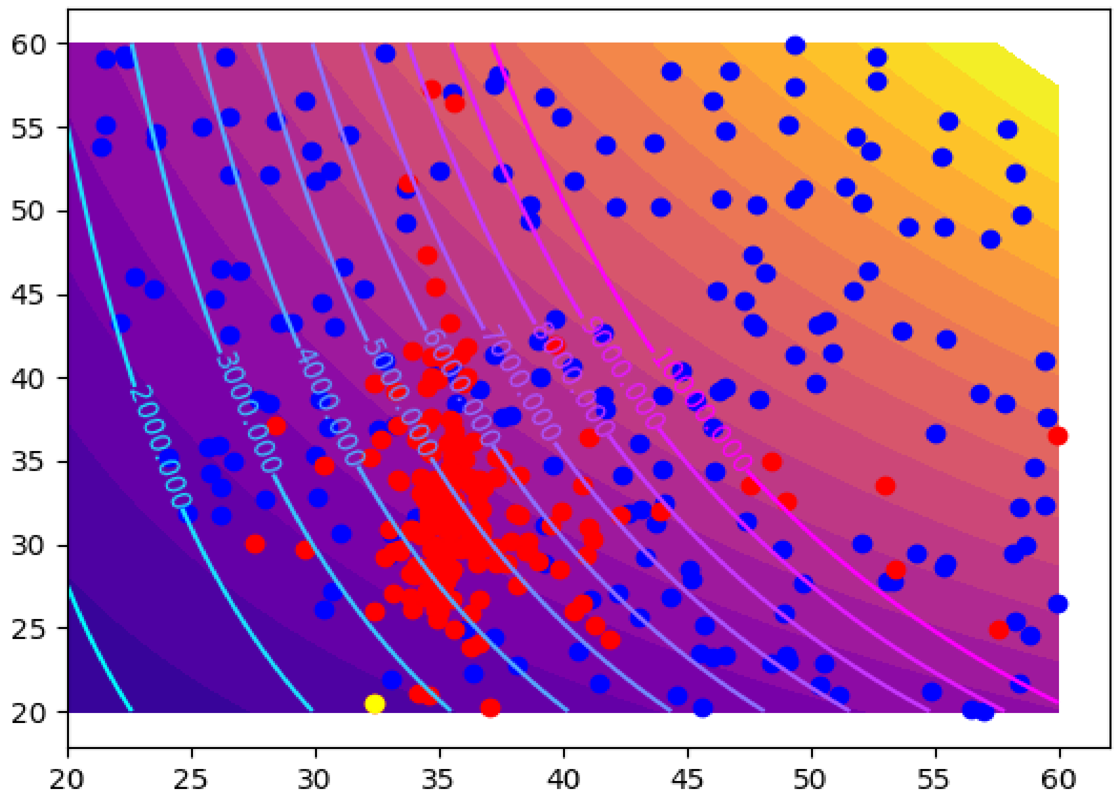

Figure 2: Solution space (x-axis = h, y-axis=b). Color gradient corresponds to cost. Isolines are breaking bending moments.